



UNIVERSIDAD  
POLITECNICA  
DE VALENCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

## ESCOLA TÈCNICA SUPERIOR D'ENGINYERIA INFORMÀTICA

### PROYECTO FIN DE CARRERA

# *“Desarrollo de un monitor de redes para un entorno Mainframe de IBM”*

Para optar a la titulación de:  
Ingeniero Técnico en Informática de Gestión

Presentado por:  
José Vicente Gómez Martínez

Dirigido por:  
Juan Carlos Casamayor Ródenas

## AGRADECIMIENTOS

Después de muchos años de haber abandonado la universidad (por aquél entonces escuela) me he decidido a realizar este proyecto fin de carrera. Distintas circunstancias personales y laborales han impedido hasta ahora que pudiera dedicarme a este menester. Recuerdo aquellos años de carrera como años de estudio intenso, pero también guardo gratos recuerdos de profesores y compañeros que se convirtieron en amigos.

Quiero darle las gracias en especial a mi mujer Tere y a mis hijas Beatriz y María porque sin su comprensión y apoyo no me hubiera resultado fácil la consecución de mi objetivo, a mi compañero y amigo Daniel por todos los momentos que compartimos diariamente y a mi director de proyecto sin el cuál este trabajo no hubiera sido posible.

Gracias.

## RESUMEN

Este proyecto consiste en el diseño, desarrollo e implantación de un sistema de monitorización para un entorno de redes mainframe de IBM. Proporciona información relativa del estado de las comunicaciones de las distintas particiones a los Departamentos de Explotación y Sistemas con el fin de garantizar su correcto uso y disponibilidad.

Entre los requisitos que debe cumplir está el de ofrecer un interfaz de usuario amigable e intuitiva, de forma que el manejo de la misma no presente ninguna complicación.

Se trata de una aplicación web desarrollada sobre un entorno Linux, Apache, Oracle y PHP que estará instalada en un servidor central accesible por todos los usuarios. El acceso al sistema de monitorización permite controlar los siguientes aspectos:

- Estado de los stack's TCP/IP.
- Estado de las VIPAS estáticas.
- Estado de los interfaces físicos.
- Estado y utilización de las VIPAS dinámicas
- Estado de las conexiones.
- Estado de los listeners.

Además de toda la programación, este proyecto incluye todos los scripts necesarios tanto para la extracción via SNMP de la información de los distintos sistemas, como los procedimientos para el mantenimiento de las tablas almacenadas en el sistema gestor de base de datos.

Aunque se trata de una herramienta de monitorización on-line también se le ha dotado de un sistema de historificación con el fin de ayudar al análisis posterior de las incidencias reportadas. Dicho sistema de historificación se presenta tanto de forma gráfica como de forma tabular.

El monitor es accesible desde cualquier terminal conectado a la red propia que disponga de un navegador instalado

## ABSTRACT

This project consists in the design, development and implantation of an online monitoring system for mainframe IBM networks. It provides relative information about the state of the communications of the different IBM partitions to the Departments of Exploitation and Systems in order to guarantee his correct use and availability.

Between the requirements that it must fulfill it is to offer one intuitive user's amicably interface, so that the managing of the same one does not present any complication.

This is a web application developed on a Linux, Apache, Oracle and PHP environment that will be installed in a central server accessible by all the users.

The access to the monitoring system allows to control the following aspects:

- State of the TCP/IP stack's.
- State of the static VIPAS.
- State of the physical interfaces.
- State and utilization of the dynamic VIPAS
- State of the connections.
- State of the listeners.

Besides the whole programming, this project includes all the necessary scripts so much for the extraction of the information of the different systems through SNMP, as the procedures for the maintenance of the tables stored in the Database System.

Though it is an on-line monitoring tool it has also been provided with an historical system in order to help to the later analysis of the brought incidents. The above mentioned historical system appears both of graphical form and of tables form.

The monitor is accessible from any terminal connected to the private network that has an installed browser.

## INDICE

### CAPÍTULO I. DESCRIPCIÓN DEL PROYECTO ..... - 1 -

I.1. Introducción .....	- 1 -
I.2. Motivación.....	- 2 -
I.3. Objetivos del proyecto .....	- 3 -
I.3.1 Objetivos Generales.....	- 3 -
I.3.2 Objetivos de la Aplicación .....	- 3 -
I.4. Alcance del proyecto.....	- 5 -
I.5. Tipología de Usuarios .....	- 5 -
I.6. Organización y Funciones Empresariales.....	- 7 -

### CAPÍTULO II. IDENTIFICACIÓN DE REQUISITOS ..... - 9 -

II.1. Introducción .....	- 9 -
II.2. Descripción de las categorías .....	- 9 -
II.3. Relación de Requisitos .....	- 9 -
II.3.1.Requisitos Funcionales .....	- 11 -
II.3.2.Requisitos Operativos .....	- 15 -
II.3.3.Requisitos de Prestaciones.....	- 24 -
II.3.4.Requisitos de Seguridad .....	- 27 -

### CAPÍTULO III. DISEÑO DE LA ARQUITECTURA ..... - 28 -

III.1. Introducción .....	- 28 -
III.2. Tipo de Arquitectura.....	- 28 -
III.3. Solución Hardware.....	- 31 -
III.4. Solución Software .....	- 34 -

### CAPÍTULO IV. MODELO DE DATOS ..... - 35 -

IV.1. Introducción .....	- 35 -
IV.2. Modelo Lógico .....	- 39 -

IV.2.1. DFD contextual. ....	- 39 -
IV.2.2. DFD conceptual de primer nivel.....	- 40 -
IV.2.3. DFD conceptual de segundo nivel. EXPL. VTL3, CEIN.....	- 44 -
IV.2.4. DFD conceptual de segundo nivel. SIAC, SIAB.....	- 47 -
IV.3. Modelo conceptual de datos.....	- 51 -
IV.3.1. Modelo entidad-relacion.....	- 51 -
IV.4. Modelo físico. ....	- 52 -
IV.4.1. Definiciones de la Base de Datos. ....	- 52 -
<b>CAPÍTULO V. PROGRAMACION .....</b>	<b>- 112 -</b>
V.1. Introducción.....	- 112 -
V.2 Obtención de Datos.....	- 120 -
V.3 Carga de Datos .....	- 124 -
V.4 Presentación de los Datos (Diagrama de flujo) .....	- 135 -
V.5 Programas.....	- 136 -
<b>CAPITULO VI. MANUAL DE USUARIO .....</b>	<b>- 139 -</b>
<b>CAPITULO VII. PLANIFICACION TEMPORAL .....</b>	<b>- 144 -</b>
<b>CAPITULO VIII.BIBLIOGRAFIA .....</b>	<b>- 145 -</b>

## CAPÍTULO I. DESCRIPCIÓN DEL PROYECTO

### I.1. Introducción

Un sistema de información, de una magnitud y dimensión considerables, debe llevar asociado una herramienta que permita monitorizar con un nivel preciso de detalle, todos aquellos aspectos que derivan de la explotación del mismo. Esta afirmación cobra aún mayor trascendencia cuando el sistema a auditar en cuestión es la red de una entidad bancaria, tal y como representa BANCAJA.

Bancaja (Caja de Ahorros de Valencia, Castellón y Alicante) es una Entidad de Crédito de naturaleza fundacional y carácter benéfico-social, que tiene por objeto contribuir a la consecución de los intereses generales, mediante el desarrollo económico y social en su zona de actuación.

El sistema de información de BANCAJA está conformado en su parte central por dos mainframes de IBM. Los mainframes o sistemas host son máquinas cuya principal tarea es el almacenado, proceso y transferencia de grandes volúmenes de información, permitiendo el acceso compartido por varios usuarios al mismo sistema de una manera eficiente. Los sistemas mainframe se caracterizan por su alta disponibilidad, mantenibilidad (los recursos no dejan de estar disponibles por el hecho de realizar tareas de mantenimiento), capacidad de ejecución, escalabilidad y seguridad de datos. Los mainframes funcionan bajo sistema operativo IBM z/OS y utilizan como software central de comunicaciones el VTAM. Si bien el VTAM ha sido típicamente asociado a las comunicaciones SNA, lo cierto es que hoy en día la mayoría de las instalaciones han realizado una paulatina migración a comunicaciones TCP/IP más orientadas a aplicaciones cliente-servidor. Las comunicaciones se han convertido en una parte fundamental, si no la más importante, de los actuales sistemas de información.

La necesidad de monitorizar dichas comunicaciones en aras de garantizar su correcto uso y disponibilidad, así como poder descubrir y depurar posibles errores o situaciones anómalas y mantener con ello la calidad de los mismas, me induce a desarrollar el Monitor de Redes para el entorno Mainframe de BANCAJA.

## I.2. Motivación

Actualmente BANCAJA no dispone de un sistema eficaz de monitorización de las comunicaciones TCP/IP de sus entornos IBM, por lo que el control de los mismos se hace complicado y farragoso sobre todo para el departamento de Explotación que es el encargado de velar por la disponibilidad del sistema 24 horas al día 365 días al año.

La principal motivación para el desarrollo de este proyecto está en que el fabricante no dispone de una herramienta específica que permita visualizar el conjunto de elementos que lo componen, así como su estado en cada momento.

La segunda motivación está en la aplicación de los conocimientos adquiridos durante la carrera y sobre todo durante mi dilatada experiencia profesional en diversas áreas de la informática. Un proyecto de estas características hace uso de conocimientos en sistemas operativos, comunicaciones, programación, bases de datos, seguridad informática, estructuras de datos, ingeniería del software, gestión de proyectos y gestión de sistemas.



### I.3. Objetivos del proyecto

A continuación vamos a describir los objetivos que se pretende conseguir con con el desarrollo de este proyecto.

#### I.3.1 Objetivos Generales

1. Centralización de toda la información de monitorización de los mainframes en un único punto permitiendo que sea accesible para los integrantes de los distintos departamentos.
2. Asegurar la confidencialidad de la información mediante políticas de seguridad que garanticen que los datos sólo son accedidos por quien tiene permiso para hacerlo.
3. Garantizar la disponibilidad de la información. Se utilizará un sistema de alta disponibilidad con servidores redundantes.
4. Diseñar una interfaz que sea amigable e intuitiva, disminuyendo así la curva de aprendizaje de los usuarios del sistema.
5. Accesibilidad de la información desde cualquier punto.
6. Exportación de datos a otros formatos.

#### I.3.2 Objetivos de la Aplicación

1. **Rápido despliegue de la aplicación:** Al ser una aplicación web realizada bajo el modelo cliente-servidor no hace falta instalar software extra en los equipos, basta con tener un navegador web. Este sistema evita que en cada equipo haya que instalar la aplicación y resuelve los problemas de incompatibilidad de librerías, sistemas operativos, etc.
2. **Usabilidad:** Los usuarios ya tienen experiencia en navegar por la web por lo que el uso de esta aplicación no les va a costar trabajo. Aún así se ha hecho un esfuerzo adicional de forma que la aplicación tenga el mismo aspecto que el resto de monitores de la instalación.

3. **Fiabilidad de los datos:** Toda la información se guarda en un gestor de base de datos que mantiene la coherencia de la información.
4. **Seguridad:** La seguridad es un aspecto muy importante, por ello se han implantado distintas medidas de seguridad en el gestor de base de datos, en el servidor web, a nivel de aplicación e incluso en los perfiles de los usuarios que accederán al MLT.
5. **Economía:** Todo el sistema esta montado primando la reutilización de los recursos ya existentes.
6. **Facilidad de actualización:** Un entorno web permite facilitar la actualización de la aplicación ya que los cambios se aplican en un sólo punto y quedan reflejados para todos los usuarios.
7. **Representación gráfica de datos:** No sólo se deben monitoricar los datos online sino que se deberá poder representar datos históricos mediante gráficas para la realización de informes de incidencias.

#### I.4. Alcance del proyecto

El proyecto de Desarrollo de un Monitor de redes para un entorno Mainframe de IBM dará como resultado el producto MLT que cumplirá con todos los objetivos especificados en el apartado anterior.

En el MLT se incluirán únicamente las particiones de Producción dejándose el resto de particiones (particiones de Desarrollo y TEST) fuera de éste por motivos de coherencia de la información. No se descarta la posibilidad de desarrollar un monitor adhoc para estas últimas utilizando la misma tecnología que el MLT.

El MLT será únicamente una herramienta de visualización de los estados de los componentes del sistema, no se podrá dar comandos para la resolución de incidencias, dejándose esta labor a otras herramientas propietarias de IBM como el Netview, o las propias consolas del Sistema.

#### I.5. Tipología de Usuarios

Existen distintos perfiles de usuario a los que va dirigido el nuevo monitor, cada uno con una problemática diferente. Los usuarios que van a utilizar principalmente el monitor son los siguientes:

##### **Jefe de Turno**

##### **Descripción:**

Asignada al departamento de Explotacion es la persona encargada de hacer cumplir normas y procedimientos para mantener la disponibilidad del sistema informático (mantenimiento preventivo y correctivo). También supervisa la instalación de nuevo hardware y software asociado.

##### **Obligaciones y Responsabilidades:**

Definir y hacer cumplir el esquema de mantenimiento para los diversos equipos. Implementar mecanismos para solucionar problemas no previstos en el hardware y software. Establecer procedimientos para controlar instalación por parte de terceros. Elaborar reportes e informes de fallos y necesidades.

**Se relaciona con:**

Jefe de Explotación, Planificador, Personal de Sistemas.

## Planificador

**Descripción:**

Se encarga de planificar, ejecutar y verificar los procesos, programas y resultados diarios de la instalación. Comprueba que los datos provenientes de distintas vías se incorporarán convenientemente en el sistema.

**Obligaciones y Responsabilidades:**

Operar dentro de los formatos apropiados en relación al ingreso de datos. Detectar errores y duplicaciones, realizar las operaciones rutinarias básicas de un centro de proceso de datos (procedimientos de backup, consolidaciones, etc).

**Se relaciona con:**

Jefe de Explotación, Jefe de Turno, Personal de Sistemas.

## Técnico de Sistemas Comunicaciones

**Descripción:**

Asignado al departamento de Técnica de Sistemas supervisa la operación del hardware y software relacionado con las comunicaciones. Participa en la selección de equipamiento y supervisa el mantenimiento y eficiencia de las redes de comunicación.

**Obligaciones y Responsabilidades:**

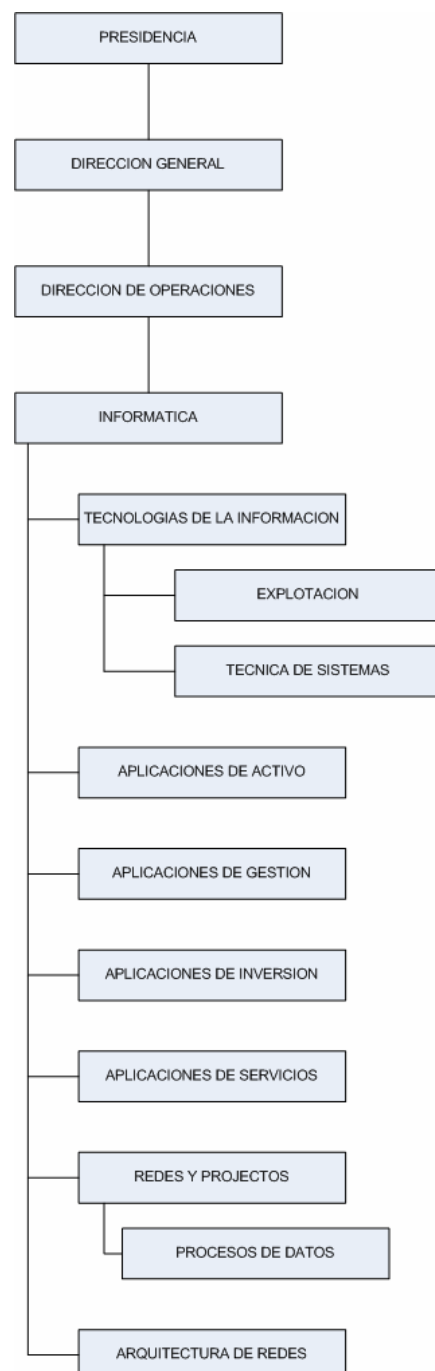
Diseña el plan básico de las redes de comunicaciones. Supervisa el rendimiento de las redes. Capacita al personal de operaciones y usuarios en relación al funcionamiento de las telecomunicaciones. Evalúa posibles mejoras en las redes de comunicaciones.

**Se relaciona con:**

Todos los usuarios y personal de Explotación, Director de Sistemas, Proveedores de hardware o software relacionado con las comunicaciones.

## I.6. Organización y Funciones Empresariales

En este punto se presenta el organigrama de las partes implicadas en el proyecto. En el proyecto existe un único actor implicado en la utilización del mismo que será el departamento de Tecnologías de la Información, pero dentro de este departamento se subdivide en otras 2 ramas: Técnica de Sistemas y Explotación, siendo los empleados de esta rama los que tendrán una implicación más directa con el proyecto.



#### Explotación:

El Jefe de Turno junto con el Planificador, componentes de esta ramificación del departamento, son los clientes más importante de este proyecto y serán quienes determinen los objetivos del mismo. Para profundizar en el proceso se realizan entrevistas con el resto de los empleados de este departamento quienes, al fin y al cabo, serán los principales usuarios finales de la aplicación.

#### Técnica de Sistemas:

El acceso a la aplicación desde esta ramificación será principalmente de los Técnicos de Sistemas de Comunicaciones.

## CAPÍTULO II. IDENTIFICACIÓN DE REQUISITOS

### II.1. Introducción

A continuación, se exponen los requisitos que debe cumplir la aplicación. Para ello se han tenido en cuenta la información extraída a partir de los diferentes encuentros y entrevistas que se han tenido con los usuarios. Se han recogido en modo de fichas individuales, fácilmente identificables y que ayudarán al diseño del nuevo sistema.

### II.2. Descripción de las categorías

Es conveniente hacer una división de requisitos atendiendo a la naturaleza de los mismos. Así, se pueden encontrar diferentes tipos de requisitos:

- Funcionales:

Atienden a características propias de las funciones de negocio (R01).

- Operativos:

Atienden al modo en que operará el sistema (R02).

- De prestaciones:

Atienden a características adicionales o funciones de menor prioridad (R03).

- De seguridad:

Atienden al control del acceso al sistema y la privacidad de los datos (R04).

- De fiabilidad:

Atienden a la integridad y veracidad de la información (R05).

### II.3. Relación de Requisitos

#### Funcionales

- Interfaz Gráfico Web.
- Presentación estándar de la instalación.
- Presentación de la información compartimentada.
- Información histórica de los elementos monitorizados.

## Operativos

- Aprovechar los recursos de software y hardware de la empresa.
- Acceso desde el navegador de la plataforma.
- Tiempo de búsquedas Base de Datos óptimo.
- Estado general del stack.
- Estado de los interfaces lógicos y físicos.
- Estado de las VIPA.
- Estado de las conexiones con SERMEPA.
- Estado de los listeners.
- Hora de recepción de los datos.

## De Prestaciones

- Sistema de Alarmas.
- Barra de desplazamiento en el caso de los listeners.
- Obtención de gráficos.

## De Seguridad

- Acceso mediante contraseñas seguras.



### II.3.1.Requisitos Funcionales

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
Requisitos			
Título	Interfaz Gráfico Web.		
Identificador	R01.1	Fecha	02/03/2010
Versión	1.0	Prioridad	Alta
Estado	Aceptado	Categoría	Funcional
Descripción	La herramienta dispondrá de un interfaz gráfico web para su explotación.		
Control			
Se creará una herramienta gráfica web tanto para la monitorización como para el tratamiento de toda la información relacionada.			
Beneficios			
El usuario evita la utilización de comandos farragosos así como la necesidad de múltiples accesos para obtener la misma información.			
Comentarios/Soluciones sugeridas			
Ninguno			
Documentos relacionados			
Ninguno			
Requisitos relacionados			
Ninguno			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Presentación estándar de la instalación		
<u>Identificador</u>	R01.2	<u>Fecha</u>	03/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Alta
<u>Estado</u>	Aceptado	<u>Categoría</u>	Funcional
<u>Descripción</u>	El monitor deberá ajustarse a las características estándar del resto de los monitores de la instalación en cuanto a presentación y usabilidad.		
<u>Control</u>			
Se utilizará el formato, las cabeceras, aspecto general, código de colores así como los controles estándar de los monitores de la instalación.			
<u>Beneficios</u>			
La adaptación del usuario al nuevo monitor se realiza de forma muy rápida e intuitiva al estar completamente adaptado al entorno.			
<u>Comentarios/Soluciones sugeridas</u>			
Ninguno			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
R01.1			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Presentación de la información compartimentada		
<u>Identificador</u>	R01.3	<u>Fecha</u>	03/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Alta
<u>Estado</u>	Aceptado	<u>Categoría</u>	Funcional
<u>Descripción</u>	El monitor deberá presentar toda la información compartimentada para su correcta interpretación por parte de los usuarios.		
<u>Control</u>			
Se presentará la información perfectamente compartimentada por partición, stack, interfaces físicos, interfaces virtuales, sermepa y listeners.			
<u>Beneficios</u>			
La interpretación de los datos por parte de los usuarios se hará de una forma más rápida e intuitiva, por lo que el usuario conocerá de un solo vistazo el estado de todo el sistema.			
<u>Comentarios/Soluciones sugeridas</u>			
El orden de las particiones se corresponderá con su ubicación física (EXPL, VTL3, SIAC, SIAB,CEIN).			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
Ninguno			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Información histórica de los elementos monitorizados		
<u>Identificador</u>	R01.4	<u>Fecha</u>	04/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Media
<u>Estado</u>	Aceptado	<u>Categoría</u>	Funcional
<u>Descripción</u>	El monitor deberá tener la posibilidad de obtener información histórica de los elementos representados para su posible consulta por parte de los usuarios.		
<u>Control</u>			
Se presentará información histórica tanto de forma gráfica como de forma tabular en todos los elementos que se controlen y así lo permitan.			
<u>Beneficios</u>			
La consulta de datos históricos sin tener que ser de una profundidad excesiva ayuda al análisis de problemas y por lo tanto redundan en una mayor disponibilidad de todo el sistema.			
<u>Comentarios/Soluciones sugeridas</u>			
La profundidad de los datos históricos se establece en un máximo de 3 días por elemento.			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
Ninguno			

### II.3.2.Requisitos Operativos

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
Requisitos			
Título	Aprovechar los recursos de software y hardware de la empresa		
Identificador	R02.5	Fecha	04/03/2010
Versión	1.0	Prioridad	Alta
Estado	Aceptado	Categoría	Operativo
Descripción	Se desea aprovechar los recursos de la empresa al máximo minimizando la inversión.		
Control			
Se intentará por todo los medios que el desembolso tanto en hardware como en software sea mínimo.			
Beneficios			
Reducción de la inversión.			
Comentarios/Soluciones sugeridas			
Ninguno			
Documentos relacionados			
Ninguno			
Requisitos relacionados			
Ninguno			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Acceso desde el navegador de la plataforma.		
<u>Identificador</u>	R02.6	<u>Fecha</u>	05/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Media
<u>Estado</u>	Aceptado	<u>Categoría</u>	Operativo
<u>Descripción</u>	El monitor deberá ser accesible desde cualquier PC de la instalación.		
<u>Control</u>			
Se accederá al monitor desde cualquier navegador instalado en cualquier PC de la instalación que tenga instalada la plataforma estándar no importando su ubicación física.			
<u>Beneficios</u>			
Independientemente de como y dónde se conecte, el usuario autorizado dispone del mismo acceso.			
<u>Comentarios/Soluciones sugeridas</u>			
Ninguno			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
R01.1			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Tiempo de búsquedas Base de Datos óptimo.		
<u>Identificador</u>	R02.7	<u>Fecha</u>	08/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Media
<u>Estado</u>	Aceptado	<u>Categoría</u>	Operativo
<u>Descripción</u>	El tiempo de acceso a las distintas tablas del Sistema Gestor de Base de Datos deberá estar muy optimizado dada la naturaleza (online) y el objetivo del monitor .		
<u>Control</u>			
El tiempo de respuesta de los accesos al SGBD deberá ser inferior a los 5 segundos.			
<u>Beneficios</u>			
El usuario obtendrá la información en un tiempo óptimo redundando en la confianza de los datos presentados.			
<u>Comentarios/Soluciones sugeridas</u>			
Ninguno			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
Ninguno			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
Requisitos			
Título	Estado general del stack.		
Identificador	R02.8	Fecha	08/03/2010
Versión	1.0	Prioridad	Alta
Estado	Aceptado	Categoría	Operativo
Descripción	El estado general de cada uno de los stack's individualmente se basará en el número de sockets establecidos en cada intervalo de medición.		
Control			
Se representará por cada uno de los stack's el número de sockets establecidos, la hora del intervalo de refresco y el delta de número de sockets establecidos en el intervalo con el fin de comprobar el correcto funcionamiento tanto del propio stack como del monitor en sí mismo.			
Beneficios			
El usuario puede comprobar si el número de sockets es correcto y si se incrementa.			
Comentarios/Soluciones sugeridas			
En el caso de que no se incremente o no se actualice se deberá activar el sistema de alarmas.			
Documentos relacionados			
Ninguno			
Requisitos relacionados			
R01.3			



Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
Requisitos			
Título	Estado de los interfaces lógicos y físicos.		
Identificador	R02.9	Fecha	08/03/2010
Versión	1.0	Prioridad	Alta
Estado	Aceptado	Categoría	Operativo
Descripción	Se presentará tanto el estado de los interfaces lógicos como de los interfaces físicos siguiendo el código de colores estándar de la instalación.		
Control			
Se visualizará en verde si el interface (lógico o físico) está operativo. En cualquier otro caso se utilizará el código de colores diseñado para tal efecto y se activará el sistema de alarmas.			
Beneficios			
El usuario sabe de forma automática cual es el estado de todos los interfaces monitorizados y la acción que tiene que realizar en el caso de que sea necesario actuar.			
Comentarios/Soluciones sugeridas			
Ninguno			
Documentos relacionados			
Ninguno			
Requisitos relacionados			
R01.2			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Estado de las VIPA.		
<u>Identificador</u>	R02.10	<u>Fecha</u>	08/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Alta
<u>Estado</u>	Aceptado	<u>Categoría</u>	Operativo
<u>Descripción</u>	El monitor deberá presentar cual es el estado de las VIPA así como de su utilización.		
<u>Control</u>			
Se representará el estado y la ubicación de las VIPA utilizando el código de colores estándar de la instalación, además se representará el delta de conexiones que están atendiendo cada uno de ellas en el intervalo.			
<u>Beneficios</u>			
El usuario sabe de forma automática cual es el estado las VIPA y la acción que tiene que realizar en el caso de que sea necesario actuar.			
<u>Comentarios/Soluciones sugeridas</u>			
Ninguno			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
R01.2			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Estado de las conexiones con SERMEPA.		
<u>Identificador</u>	R02.11	<u>Fecha</u>	08/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Alta
<u>Estado</u>	Aceptado	<u>Categoría</u>	Operativo
<u>Descripción</u>	El monitor deberá presentar cual es el estado de las distintas conexiones con SERMEPA así como la partición donde están operativas.		
<u>Control</u>			
Se representará el estado y la ubicación de cada una de las 5 conexiones que se tiene con SERMEPA utilizando el código de colores estándar de la instalación.			
<u>Beneficios</u>			
El usuario sabe de forma automática cual es el estado de todas las conexiones con SERMEPA y la acción que tiene que realizar en el caso de que sea necesario actuar.			
<u>Comentarios/Soluciones sugeridas</u>			
Ninguno			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
R01.2			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Estado de los listeners.		
<u>Identificador</u>	R02.12	<u>Fecha</u>	10/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Alta
<u>Estado</u>	Aceptado	<u>Categoría</u>	Operativo
<u>Descripción</u>	El monitor deberá presentar cual es el estado de los listeners para cada una de las particiones.		
<u>Control</u>			
Se representará el estado de los listeners, en el caso de que cualquiera de los listeners monitorizados no se encuentre activo se pondrá en rojo, se pasará al principio de la lista y se activará el sistema de alarmas.			
<u>Beneficios</u>			
El usuario sabe de forma automática cual es el estado de todas los listeners.			
<u>Comentarios/Soluciones sugeridas</u>			
Ninguno			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
Ninguno			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Hora de recepción de los datos.		
<u>Identificador</u>	R02.13	<u>Fecha</u>	12/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Media
<u>Estado</u>	Aceptado	<u>Categoría</u>	Operativo
<u>Descripción</u>	El monitor deberá presentar la hora de recepción de los datos individualmente para cada uno de los stack's.		
<u>Control</u>			
Se representará la hora de recepción de los datos de forma individualizada para cada uno de los stack's con el fin de detectar cualquier anomalía en cualquiera de los procesos que atañen al monitor (extracción de datos, carga de datos, presentación de los datos). Cuando la hora de recepción supere el tiempo equivalente a 2 muestras (8 minutos) se pondrá en rojo y se activará el sistema de alarmas.			
<u>Beneficios</u>			
El usuario sabe en todo momento que los datos que se están representando se están actualizando correctamente.			
<u>Comentarios/Soluciones sugeridas</u>			
Ninguno			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
Ninguno			

### II.3.3.Requisitos de Prestaciones

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Sistema de Alarmas.		
<u>Identificador</u>	R03.14	<u>Fecha</u>	12/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Alta
<u>Estado</u>	Aceptado	<u>Categoría</u>	Prestaciones
<u>Descripción</u>	El monitor deberá implementar un sistema de alarmas tanto visuales como sonoras con el fin de denunciar cualquier anomalía.		
<u>Control</u>			
Se implementará un sistema de alarmas visual y sonoro por partición, fácilmente desactivable, con el fin de denunciar cualquier anomalía que se presente en cualquiera de los elementos monitorizados (stack, interfaces lógicos, interfaces físicos, VIPAS, conexiones SERMEPA, listeners o fallo en la monitorización).			
<u>Beneficios</u>			
El usuario no tiene que estar constantemente atendiendo al monitor, puesto que cualquier anomalía se anuncia visual y sonoramente, informando adecuadamente de los incidentes.			
<u>Comentarios/Soluciones sugeridas</u>			
Ninguno			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
Ninguno			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Barra de desplazamiento en el caso de los listeners.		
<u>Identificador</u>	R03.15	<u>Fecha</u>	12/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Media
<u>Estado</u>	Aceptado	<u>Categoría</u>	Prestaciones
<u>Descripción</u>	El monitor deberá posibilitar el desplazamiento en la representación de los listeners.		
<u>Control</u>			
Debido al alto y cambiante número de listeners en algunas particiones se hace imprescindible la utilización de una barra de desplazamiento para poder acceder a los listeners ocultos.			
<u>Beneficios</u>			
El usuario puede acceder de una forma sencilla a la totalidad de los listeners.			
<u>Comentarios/Soluciones sugeridas</u>			
Se procederá a la reordenación automática de los mismos ante incidentes, con el fin de que estos aparezcan los primeros.			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
Ninguno			

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Obtención de gráficos.		
<u>Identificador</u>	R03.16	<u>Fecha</u>	12/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Baja
<u>Estado</u>	Aceptado	<u>Categoría</u>	Prestaciones
<u>Descripción</u>	El monitor deberá posibilitar la obtención de gráficos históricos y tabulares para la presentación de informes ante incidentes.		
<u>Control</u>			
Cuando accedamos a los elementos monitorizados se obtendrán gráficos y tablas históricos de los mismos.			
<u>Beneficios</u>			
El usuario puede acceder a información histórica para la documentación de incidentes.			
<u>Comentarios/Soluciones sugeridas</u>			
Ninguno			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
Ninguno			



### II.3.4.Requisitos de Seguridad

Proyecto: Desarrollo de un monitor de redes para un entorno Mainframe de IBM			
<u>Requisitos</u>			
<u>Título</u>	Acceso mediante contraseñas seguras.		
<u>Identificador</u>	R04.17	<u>Fecha</u>	12/03/2010
<u>Versión</u>	1.0	<u>Prioridad</u>	Media
<u>Estado</u>	Aceptado	<u>Categoría</u>	Seguridad
<u>Descripción</u>	El acceso al monitor se protegerá mediante usuario/password.		
<u>Control</u>			
El acceso al monitor se realizará solo por los usuarios autorizados, a pesar de que no existe ningún riesgo operativo para la instalación puesto que desde el no se puede interactuar con los sistemas.			
<u>Beneficios</u>			
Impedir acceso a los usuarios no autorizados.			
<u>Comentarios/Soluciones sugeridas</u>			
Ninguno			
<u>Documentos relacionados</u>			
Ninguno			
<u>Requisitos relacionados</u>			
Ninguno			

## CAPÍTULO III. DISEÑO DE LA ARQUITECTURA

### III.1. Introducción

Bancaja históricamente siempre ha estado muy concienciada con el tema de la monitorización de tal forma que cuando me planteo el desarrollo de este proyecto ya existe una infraestructura hardware y software habilitada para tal efecto. La existencia de dicha infraestructura plenamente asentada y explotada plantea una gran cantidad de ventajas y muy pocos inconvenientes. Voy proceder a enumerar dichas ventajas e inconvenientes.

#### Ventajas

- Solución Hardware redundante instalada y probada.
- Personal especializado encargado de su mantenimiento y control.
- Reutilización de recursos hardware, software base y personal .
- Rápida implantación del proyecto.
- Solución software base disponible con modificaciones mínimas.

#### Inconvenientes

- Imposición del diseño.
- Sincronización de cambios.

### III.2. Tipo de Arquitectura

La arquitectura de la solución de monitorización BANCAJA es una arquitectura de cliente / servidor con cliente ligero. En concreto los monitores son aplicaciones web. En cuanto al sistema operativo del servidor se trata de un servidor Linux sobre el que está montado un servidor web Apache y un gestor de base de datos ORACLE. El lenguaje de programación estándar es PHP.

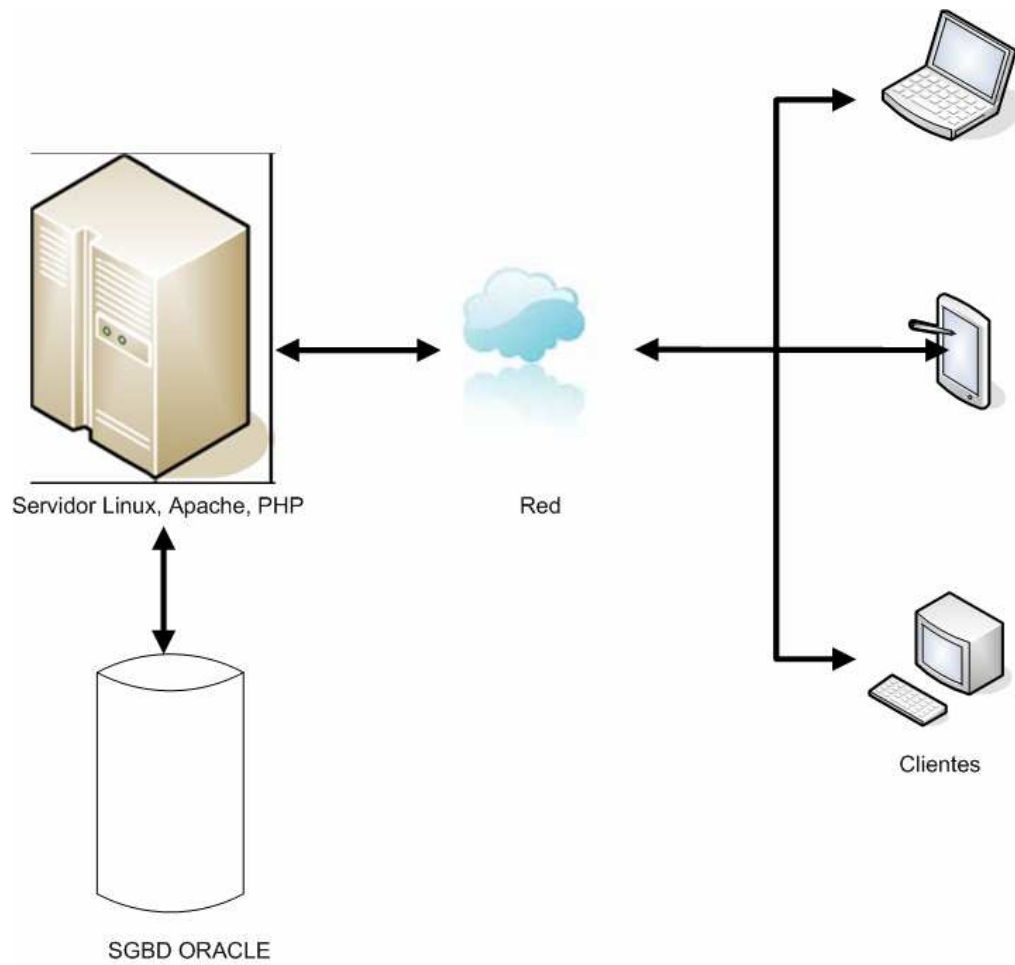
Un cliente liviano o cliente ligero (thin client o slim client) es un ordenador cliente o un software de cliente en una arquitectura de red cliente-servidor que depende primariamente del servidor central para las tareas de procesamiento, y principalmente se enfoca en transportar la entrada y la salida entre el usuario y el servidor remoto. En contraste, un cliente pesado hace tanto procesamiento como sea posible y pasa solamente los datos para las comunicaciones y el almacenamiento al servidor.

Muchos dispositivos de cliente liviano ejecutaban solamente navegadores web o programas de escritorio remoto, lo que significaba que todo el procesamiento significativo ocurría en el servidor. Sin embargo, dispositivos recientes etiquetados como clientes livianos pueden correr sistemas operativos completos tales como Linux Debian, calificándose como nodos sin disco o clientes híbridos. Algunos clientes ligeros también son llamados "terminales del acceso".

Por consecuencia, el término "cliente ligero", en términos de hardware, ha venido a abarcar cualquier dispositivo usado como, un cliente ligero en la definición original, incluso si sus capacidades reales son mucho mayores. El término también es a veces usado en un sentido incluso más amplio que incluye nodos sin disco.

En un sistema cliente liviano-servidor, el único software que es instalado en el cliente ligero es la interface de usuario, algunas aplicaciones frecuentemente usadas, y un sistema operativo de red. Este software puede ser cargado desde una unidad de disco local, del servidor en tiempo de arranque, o según lo que se necesite. Al simplificar la carga en el cliente ligero, éste puede ser un dispositivo muy pequeño y de baja energía, que da costos de compra y de operación más bajos en cada puesto. El servidor, o un cluster de servidores tiene el peso total de todas las aplicaciones, servicios, y datos. Al mantener algunos servidores ocupados y muchos clientes livianos ligeramente cargados, los usuarios pueden esperar una administración de sistemas más fácil y costos más bajos, así como todas las ventajas de la computación en red: almacenamiento y respaldo centralizados y una seguridad más fácil.

El motivo para haber elegido BANCAJA esta arquitectura web cliente / servidor está en que el acceso a los monitores se realizara desde plataformas diferentes. Una aplicación web cubre plenamente este requisito. La elección del sistema operativo Linux y el servidor web Apache cubren el requisito de utilizar software libre en la medida de lo posible. La elección de un Sistema Gestor de Base de Datos ORACLE se debe a que se van a manejar una gran cantidad de datos y este está considerado como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad, escalabilidad y su soporte multiplataforma. Finalmente se seleccionó PHP como lenguaje de programación en el lado del servidor por la fuerte integración de éste con el servidor web. Además, PHP dispone de una amplia librería extendiendo la funcionalidad del lenguaje y permitiendo que éste interactúe con el sistema operativo, servidores LDAP, etc.



### III.3. Solución Hardware

El servidor sobre el que está montado la solución tiene las siguientes características:

#### Tipo de maquina

##### System Information

Manufacturer: FUJITSU SIEMENS  
Product Name: PRIMERGY RX100S5  
Version: GS01  
Serial Number: YK2F003049  
UUID: 594B3246-3030-3330-3439-000AE4850A52  
Wake-up Type: Power Switch  
SKU Number: ABN:K1160-V201-1  
Family: SERVER

#### Procesador

##### Processor Information

Socket Designation: CPU1  
Type: Central Processor  
Family: Core 2  
Manufacturer: Intel  
ID: FB 06 00 00 FF FB EB BF  
Signature: Type 0, Family 6, Model 15, Stepping 11  
Version: Intel(R) Xeon(R) CPU 3065  
Voltage: 3.3 V  
External Clock: Unknown  
Max Speed: 2333 MHz  
Current Speed: 2333 MHz  
Status: Populated, Enabled  
Upgrade: ZIF Socket  
L1 Cache Handle: 0x0005  
L2 Cache Handle: 0x0006  
L3 Cache Handle: Not Provided  
Serial Number: Not Specified  
Asset Tag: Not Specified  
Part Number: Not Specified

#### Memoria Cache

##### Cache Information

Socket Designation: L1 Cache  
Configuration: Enabled, Not Socketed, Level 1  
Operational Mode: Write Through  
Location: Internal  
Installed Size: 64 kB  
Maximum Size: 64 kB  
Supported SRAM Types:  
    Burst  
    Synchronous  
Installed SRAM Type: Burst Synchronous  
Speed: Unknown  
Error Correction Type: Parity  
System Type: Data  
Associativity: Other

## Cache Information

Socket Designation: L2 Cache  
Configuration: Enabled, Not Socketed, Level 2  
Operational Mode: Write Back  
Location: Internal  
Installed Size: 4096 kB  
Maximum Size: 12288 kB  
Supported SRAM Types:  
    Burst  
    Pipeline Burst  
    Synchronous  
    Asynchronous  
Installed SRAM Type: Burst  
Speed: Unknown  
Error Correction Type: Unknown  
System Type: Unified  
Associativity: Other

Memoria

## Memory Device

Array Handle: 0x0022  
Error Information Handle: No Error  
Total Width: 72 bits  
Data Width: 64 bits  
Size: 2048 MB  
Form Factor: DIMM  
Set: None  
Locator: DIMM-1A  
Bank Locator: Not Specified  
Type: DDR2  
Type Detail: Synchronous  
Speed: 800 MHz  
Manufacturer: Not Specified  
Serial Number: Not Specified  
Asset Tag: Not Specified  
Part Number: Not Specified

## Memory Device

Array Handle: 0x0022  
Error Information Handle: No Error  
Total Width: 72 bits  
Data Width: 64 bits  
Size: 2048 MB  
Form Factor: DIMM  
Set: None  
Locator: DIMM-2A  
Bank Locator: Not Specified  
Type: DDR2  
Type Detail: Synchronous  
Speed: 800 MHz  
Manufacturer: Not Specified  
Serial Number: Not Specified  
Asset Tag: Not Specified  
Part Number: Not Specified

## Memory Device

Array Handle: 0x0022  
Error Information Handle: No Error  
Total Width: 72 bits

Data Width: 64 bits  
Size: 2048 MB  
Form Factor: DIMM  
Set: None  
Locator: DIMM-1B  
Bank Locator: Not Specified  
Type: DDR2  
Type Detail: Synchronous  
Speed: 800 MHz  
Manufacturer: Not Specified  
Serial Number: Not Specified  
Asset Tag: Not Specified  
Part Number: Not Specified

#### Memory Device

Array Handle: 0x0022  
Error Information Handle: No Error  
Total Width: 72 bits  
Data Width: 64 bits  
Size: 2048 MB  
Form Factor: DIMM  
Set: None  
Locator: DIMM-2B  
Bank Locator: Not Specified  
Type: DDR2  
Type Detail: Synchronous  
Speed: 800 MHz  
Manufacturer: Not Specified  
Serial Number: Not Specified  
Asset Tag: Not Specified  
Part Number: Not Specified

#### Disco

Disco /dev/sda: 146.8 GB, 146815737856 bytes  
255 heads, 63 sectors/track, 17849 cylinders  
Unidades = cilindros de 16065 \* 512 = 8225280 bytes

Disposit.	Inicio	Comienzo	Fin	Bloques	Id	Sistema
/dev/sda1	*	1	2611	20972826	fd	Linux raid
/dev/sda2		2612	4569	15727635	fd	Linux raid
/dev/sda3		4570	5874	10482412+	fd	Linux raid
/dev/sda4		5875	17849	96189187+	fd	Linux raid

Disco /dev/sdb: 146.8 GB, 146815737856 bytes  
255 heads, 63 sectors/track, 17849 cylinders  
Unidades = cilindros de 16065 \* 512 = 8225280 bytes

Disposit.	Inicio	Comienzo	Fin	Bloques	Id	Sistema
/dev/sdb1	*	1	2611	20972826	fd	Linux raid
/dev/sdb2		2612	4569	15727635	fd	Linux raid
/dev/sdb3		4570	5874	10482412+	fd	Linux raid
/dev/sdb4		5875	17849	96189187+	fd	Linux raid

Disco /dev/md0: 21.4 GB, 21476081664 bytes  
2 heads, 4 sectors/track, 5243184 cylinders  
Unidades = cilindros de 8 \* 512 = 4096 bytes

Disco /dev/md3: 98.4 GB, 98497658880 bytes  
2 heads, 4 sectors/track, 24047280 cylinders  
Unidades = cilindros de 8 \* 512 = 4096 bytes

Disco /dev/md1: 10.7 GB, 10733879296 bytes  
2 heads, 4 sectors/track, 2620576 cylinders  
Unidades = cilindros de 8 \* 512 = 4096 bytes

Disco /dev/md2: 16.1 GB, 16105013248 bytes  
2 heads, 4 sectors/track, 3931888 cylinders  
Unidades = cilindros de 8 \* 512 = 4096 bytes

### III.4. Solución Software

El software instalado en el servidor es el siguiente:

#### Sistema Operativo

System:

Linux moniwebh 2.6.18-164.6.1.el5 #1 SMP Tue Oct 27 11:28:30 EDT 2009 x86\_64

Build Date:

Feb 8 2010 12:21:30

#### Servidor Web

Apache Version:

Apache/2.2.3 (Red Hat)

Apache API Version:

20051115

#### Sistema Gestor de Base de Datos

Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 - 64bit Production

PL/SQL Release 11.2.0.1.0 – Production

CORE 11.2.0.1.0 Production

TNS for Linux: Version 11.2.0.1.0 – Production

NLSRTL Version 11.2.0.1.0 – Production

#### Lenguaje de programación

PHP Version 5.2.11



## CAPÍTULO IV. MODELO DE DATOS

### IV.1. Introducción

Aunque como comentamos en el **capítulo III Diseño de la Arquitectura** vamos a utilizar ORACLE como Sistema Gestor de Base de Datos (SGBD) vamos a describir brevemente tanto este como las posibles alternativas. Realizaremos un repaso entre las distintas opciones que tenemos en la actualidad y que podríamos instalar en una máquina con sistema operativo Linux como la nuestra.

#### Oracle

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), desarrollado por Oracle Corporation. Se considera a Oracle como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad, escalabilidad y su soporte multiplataforma.

Es un producto vendido a nivel mundial, aunque la gran potencia que tiene y su elevado precio hacen que por norma general sólo se vea en empresas muy grandes y multinacionales. Para desarrollar aplicaciones que usen Oracle se utiliza PL/SQL, un lenguaje de 5ª generación, bastante potente para tratar y gestionar la base de datos, también por norma general se suele utilizar SQL.

Oracle es sin duda uno de los mejores gestores de bases de datos existentes en el mercado, es un gestor muy robusto y que cuenta con numerosas características.

Ha sido criticada por algunos especialistas la seguridad de la plataforma, y las políticas de suministro de parches de seguridad. Aunque su dominio en el mercado de servidores empresariales ha sido casi total hasta hace poco, recientemente sufre la competencia de Microsoft SQL Server de Microsoft y de la oferta de otros RDBMS con licencia libre como PostgreSQL, MySQL o Firebird.

Los sistemas operativos (del servidor) que pueden utilizar este gestor son:

Windows, Linux Mac OS x, IBM AIX, Solaris, HP - UX, QNX, SCO Unix, Novell Netware, SGI Iris.

Para la obtención de más información puede visitar el sitio Web sobre Oracle <http://www.oracle.com>

## MySql

MySQL, es una de las bases de datos relacional de código abierto más popular en Internet. Existen distribuciones para distintas plataformas (Win32/NT, Unix/Linux), y en muchas de ellas se instala por defecto (por ejemplo en los Red Hat Linux). Actualmente es propiedad de MySQL AB (una empresa sueca), que se encarga de su desarrollo y ofrece servicios auxiliares (consultoría, soporte técnico, etc.). Aunque el software es gratuito, su uso en según que aplicaciones está sujeto a licencia de pago.

MySQL originalmente se desarrolló pensando en el manejo rápido y sencillo de grandes bases de datos. Actualmente sigue en desarrollo y el producto obtenido es rápido, confiable, fácil de manejar, ofrece conectividad desde distintas tecnologías, etc.

MySQL utiliza el paradigma de los sistemas cliente/servidor, implementando un servidor SQL multihilo que soporta conectividad con diversos programas clientes, bibliotecas, herramientas de administración e interfaces de programación. MySQL AB también proporciona una biblioteca que puede utilizarse para integrar soporte MySQL en aplicaciones propias. De esta manera se consiguen productos que proporcionan un acceso a BBDDs MySQL de manera más rápida y fácil de gestionar. Actualmente son muchas las aplicaciones de terceras partes que soportan la conectividad con MySQL (clientes de base de datos, lenguajes, etc.).

Para la descarga del software u obtención de más información puede visitar el sitio Web sobre MySQL <http://www.mysql.com>

## mSQL

mSQL (*Mini SQL*) es un gestor de bases de datos ligero, diseñado para proporcionar acceso rápido a conjuntos relativamente pequeños de datos almacenados en sistemas con poca memoria.

Implementa un subconjunto de SQL e inicialmente fue desarrollado como un proyecto académico en código abierto. Actualmente las últimas versiones estables son las numeradas como mSQL 2.X, aunque ya existen versiones Beta del nuevo mSQL 3.0.

Las primeras versiones de mSQL (mSQL 1.X) alcanzaron sus objetivos, son motores muy rápidos sobre conjuntos de datos pequeños. Debido a ello y a su carácter de código abierto, se popularizaron rápidamente en Internet. Pero estaban muy limitados

y no son la elección adecuada si se deben manejar aplicaciones con fuertes cargas de datos (más de 1 millón de registros) o que necesiten de consultas SQL complejas.

Las nuevas versiones (de 2.X en adelante) tratan de conseguir los siguientes objetivos:

- Proporcionar en operaciones simples el mismo rendimiento que con mSQL 1.X.
- Conseguir acceso rápido a bases de datos con gran cantidad de datos o accedidas mediante operaciones complejas.
- Completar la implementación con otras funcionalidades definidas en la especificación ANSI SQL.

El servidor de mSQL 2.X es capaz de servir múltiples peticiones simultáneas, para ello se utiliza un esquema basado en un proceso principal y varios procesos hijos. El proceso principal es capaz de aceptar conexiones de clientes, gestionar la distribución de trabajo entre los procesos hijos y realizar otras tareas de administración. mSQL 2.0 implementa bloqueos a nivel de tabla, que permiten lecturas compartidas y escritura exclusiva. El uso de bloqueos es el primer paso hacia el soporte de transacciones, pero hoy por hoy mSQL aún no soporta transacciones.

El desarrollo de mSQL continúa, aunque parece que el producto ha perdido algo de fuerza con respecto a su más directo competidor MySQL; es posible que haya influido el cambio en las condiciones de licencia, ahora para un uso comercial del software se exige el pago de licencia.

Para conseguir una versión del software, documentación, comprobar las condiciones de licencia, etc, puede visitarse el sitio de mSQL <http://www.hughes.com.au>

## PostgreSQL

PostgreSQL es un gestor de bases de datos Relacional-Objetual. Es uno de los SGBDR Open-Source con más "solera", la primera versión es de 1985 y actualmente se distribuye PostgreSQL 7.X. Está muy extendido, sobre todo en el mundo Unix/Linux (muchas distribuciones Linux, como Red Hat lo instalan por defecto), aunque existen versiones para plataformas Windows. Entre sus ventajas hay que comentar que soporta casi todas las construcciones SQL, incluso extensiones orientadas a objetos (como por ejemplo la posibilidad de definir tablas mediante herencia), en este sentido

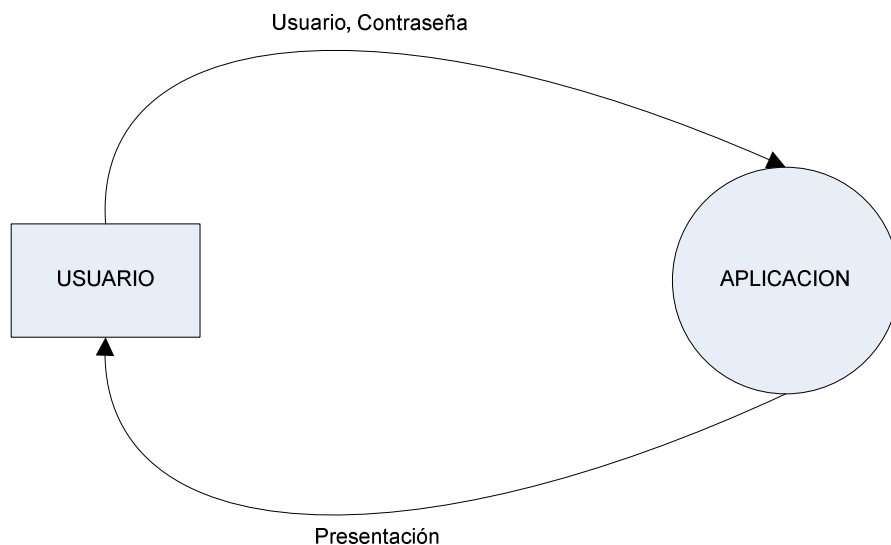
es el motor de BBDDs en código abierto más avanzado. Entre las construcciones soportadas se pueden destacar transacciones, vistas y triggers. Además dispone de una amplia conectividad a través de C, C++ y C embebido, Perl (perl5), Python (PyGreSQL), TCL (libpgtcl), PHP, JDBC y ODBC. El mayor problema que se le achaca es su lentitud frente a otras opciones en código libre como mSQL o MySQL.

Para más información puede visitarse el sitio Web de PostgreSQL <http://www.postgresql.org/>.

## IV.2. Modelo Lógico

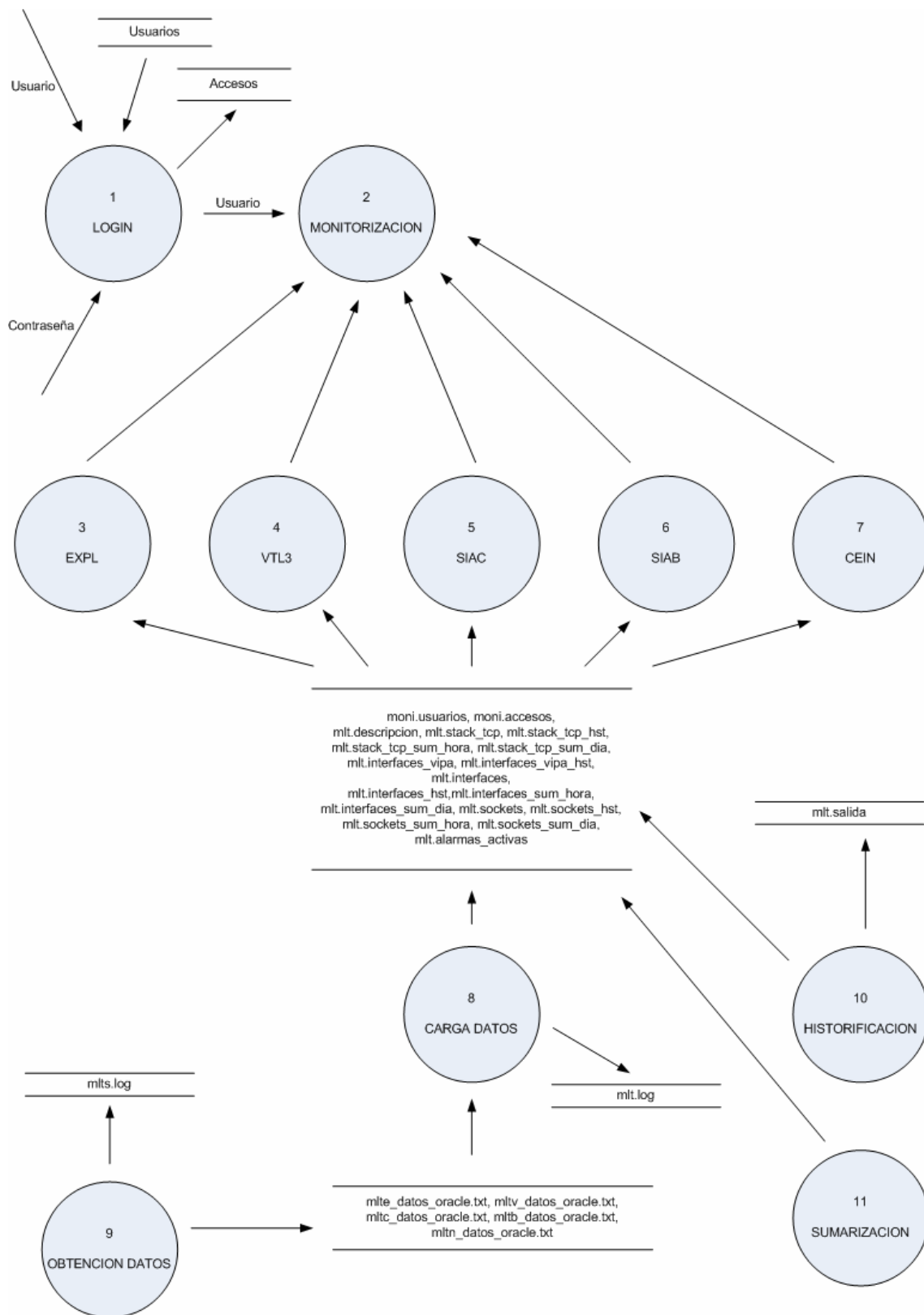
En este apartado se analiza el modelo lógico del nuevo sistema. Para su obtención se ha partido de un análisis del sistema y de la lista de requisitos del sistema.

### IV.2.1. DFD contextual.



En el diagrama se puede observar como el usuario es la única entidad externa que interactúa con la aplicación. Se considera usuario, contraseña, como los datos de autenticación, una vez autenticados el sistema presenta toda la información relativa a la monitorización del sistema. Toda la información se devuelve mediante páginas dinámicas.

### IV.2.2. DFD conceptual de primer nivel.



## MINIESPECIFICACIONES

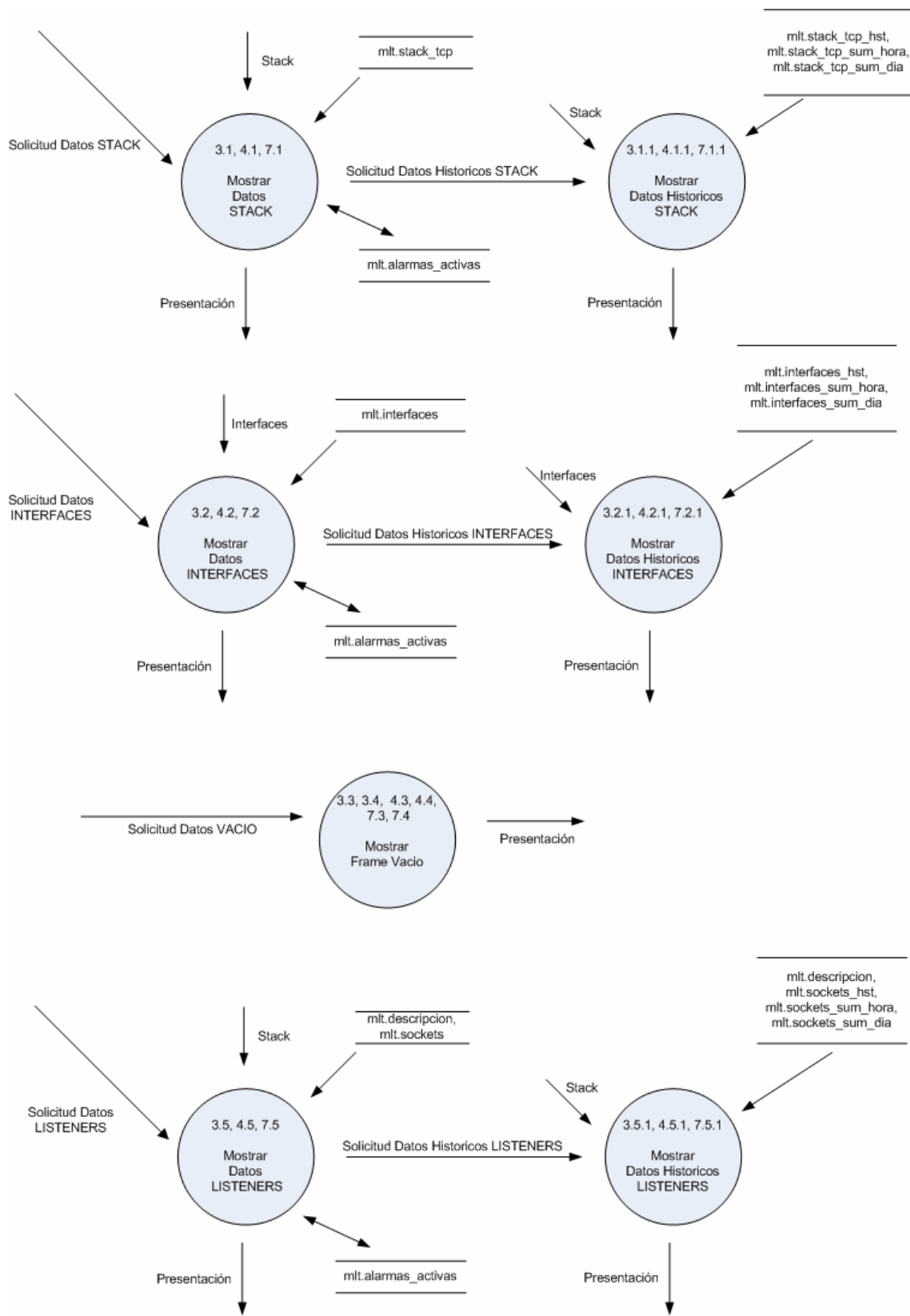
Tipo	Nombre	Atributos	Descripción
Proceso	Login	-	Proceso encargado de controlar el acceso a la aplicación.
Proceso	Monitorizacion	-	Proceso que genera la pantalla con toda la información relativa a la monitorización.
Proceso	Usuario	-	Proceso que controla la gestión de los usuarios en la aplicación. Toda actividad en este módulo queda registrada en una tabla de accesos.
Proceso	EXPL	-	Proceso que genera la información relativa a la partición de EXPL.
Proceso	VTL3	-	Proceso que genera la información relativa a la partición de VTL3.
Proceso	SIAC	-	Proceso que genera la información relativa a la partición de SIAC.
Proceso	SIAB	-	Proceso que genera la información relativa a la partición de SIAB.
Proceso	CEIN	-	Proceso que genera la información relativa a la partición de CEIN.
Proceso	Obtención Datos	-	Proceso automático que realiza las consultas a las particiones y genera los ficheros de datos.
Proceso	Carga Datos	-	Proceso automático que formatea los ficheros de datos y los carga en las tablas de la base de datos
Proceso	Historificación	-	Proceso automático de periodicidad diaria que historifica las tablas de la base de datos.
Proceso	Sumarizacion	-	Proceso ejecutado por el usuario de periodicidad mensual que suma las tablas de la base de datos.
Flujo de Datos	Usuario	-	Número de usuario. Necesario para acceder al monitor.
Flujo de Datos	Contraseña	-	Contraseña del usuario.
Almacén de Datos	moni.accesos	USUARIO, MONITOR, PERFIL	Registra las acciones realizadas con el usuario.
Almacén de Datos	moni.usuarios	USUARIO, MONITOR, FECHA, CUENTA	Información sobre los usuarios dados de alta en el sistema.
Almacén de Datos	mlt.descripcion	STACK, NOMBRE, PUERTO, DESCRIPCION, HORA_D, HORA_H, DIAS_SEM	Información sobre los listeners que se desea monitorizar en el sistema.

Almacén de Datos	mlt.stack_tcp	STACK, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.stack_tcp_hst	STACK, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.stack_tcp_sum_hora	STACK, CAMPO, FECHA, VALOR	Información sumariada por horas de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.stack_tcp_sum_dia	STACK, CAMPO, FECHA, VALOR	Información sumariada por dia de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.interfaces	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_hst	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_sum_hora	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información sumariada por horas de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_sum_dia	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información sumariada por dia de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_vipa	STACK, PUERTO_IP, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los interfaces virtuales del sistema.
Almacén de Datos	mlt.interfaces_vipa_hst	STACK, PUERTO_IP, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los interfaces virtuales del sistema.
Almacén de Datos	mlt.sockets	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los listeners y de las conexiones del sistema.
Almacén de Datos	mlt.sockets_hst	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los listeners y de las conexiones del sistema.
Almacén de Datos	mlt.sockets_sum_hora	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información sumariada por horas de los distintos parámetros de los listeners y de las conexiones del sistema.
Almacén de Datos	mlt.sockets_sum_dia	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información sumariada por dia de los distintos parámetros de los listeners y de las conexiones del sistema.
Almacén de Datos	mlt.alarmas_activas	STACK, TIPO, FECHA, ESTADO	Información de las alarmas activas en el sistema.
Almacén de Datos	mlte_datos_oracle.txt	-	Fichero con la información obtenida del sistema EXPL en formato texto.
Almacén de Datos	mltv_datos_oracle.txt	-	Fichero con la información obtenida del sistema VTL3 en formato texto.



Almacén de Datos	mltc_datos_oracle.txt	-	Fichero con la información obtenida del sistema SIAC en formato texto.
Almacén de Datos	mltb_datos_oracle.txt	-	Fichero con la información obtenida del sistema SIAB en formato texto.
Almacén de Datos	mltn_datos_oracle.txt	-	Fichero con la información obtenida del sistema CEIN en formato texto.
Almacén de Datos	mlts.log	-	Fichero con la información de errores de obtención de los datos.
Almacén de Datos	mlt.log	-	Fichero con la información de errores de carga de los ficheros.
Almacén de Datos	mlt.salida	-	Fichero con la información de errores de historificación de las tablas.

### IV.2.3. DFD conceptual de segundo nivel. EXPL. VTL3, CEIN.

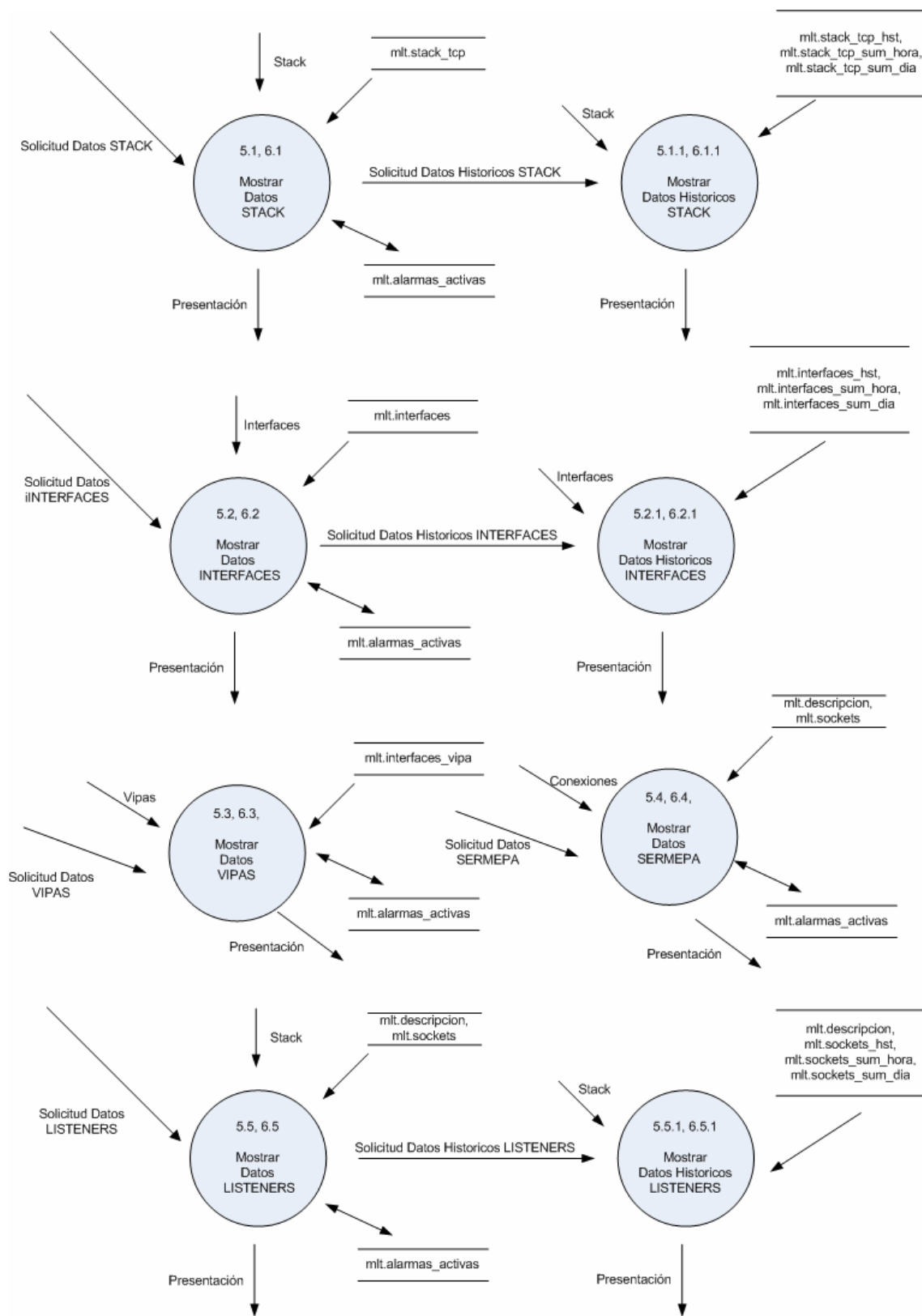


## MINIESPECIFICACIONES

Tipo	Nombre	Atributos	Descripción
Proceso	Mostrar Datos STACK	-	Proceso mediante el cual se extraen los datos del stack especificado.
Proceso	Mostrar Datos Historicos STACK	-	Proceso a petición del usuario mediante el cual se extraen los datos históricos del stack especificado.
Proceso	Mostrar Datos INTERFACES	-	Proceso mediante el cual se extraen los datos de los interfaces especificados.
Proceso	Mostrar Datos Historicos INTERFACES	-	Proceso mediante el cual se extraen los datos históricos del interface especificado.
Proceso	Mostrar Datos LISTENERS	-	Proceso mediante el cual se extraen los datos de los listeners del stack especificado.
Proceso	Mostrar Datos Historicos LISTENERS	-	Proceso mediante el cual se extraen los datos históricos del listener especificado.
Proceso	Mostrar Datos VACIO	-	Proceso mediante el cual se rellena un frame en vacio.
Proceso	Presentación	-	Proceso mediante el cual se muestran los datos extraídos.
Flujo de Datos	Stack	-	Nombre del stack del cual queremos ejecutar el proceso.
Flujo de Datos	Interfaces	-	Nombre del interface del cual queremos ejecutar el proceso.
Flujo de Datos	Solicitud Datos STACK	-	Solicitud automatizada por la que se pide la extracción de los datos del stack especificado.
Flujo de Datos	Solicitud Datos Históricos STACK	-	Solicitud a petición del usuario por la que se pide la extracción de los datos históricos del stack especificado.
Flujo de Datos	Solicitud Datos INTERFACES	-	Solicitud automatizada por la que se pide la extracción de los datos de los interfaces especificado.
Flujo de Datos	Solicitud Datos Históricos INTERFACES	-	Solicitud a petición del usuario por la que se pide la extracción de los datos históricos del interface especificado.
Flujo de Datos	Solicitud Datos LISTENERS	-	Solicitud automatizada por la que se pide la extracción de los listeners del stack especificado.
Flujo de Datos	Solicitud Datos Históricos LISTENERS	-	Solicitud a petición del usuario por la que se pide la extracción de los datos históricos del listener especificado.
Flujo de Datos	Solicitud Datos VACIO	-	Solicitud automatizada por la que se rellenan los datos en vacio.

Almacén de Datos	mlt.descripcion	STACK, NOMBRE, PUERTO, DESCRIPCION, HORA_D, HORA_H, DIAS_SEM	Información sobre los listeners que se desea monitorizar en el sistema.
Almacén de Datos	mlt.stack_tcp	STACK, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.stack_tcp_hst	STACK, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.stack_tcp_sum_hora	STACK, CAMPO, FECHA, VALOR	Información sumariada por horas de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.stack_tcp_sum_dia	STACK, CAMPO, FECHA, VALOR	Información sumariada por dia de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.interfaces	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_hst	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_sum_hora	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información sumariada por horas de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_sum_dia	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información sumariada por dia de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_vipa	STACK, PUERTO_IP, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los interfaces virtuales del sistema.
Almacén de Datos	mlt.interfaces_vipa_hst	STACK, PUERTO_IP, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los interfaces virtuales del sistema.
Almacén de Datos	mlt.sockets	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los listeners y de las conexiones del sistema.
Almacén de Datos	mlt.sockets_hst	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los listeners y de las conexiones del sistema.
Almacén de Datos	mlt.sockets_sum_hora	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información sumariada por horas de los distintos parámetros de los listeners y de las conexiones del sistema.
Almacén de Datos	mlt.sockets_sum_dia	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información sumariada por dia de los distintos parámetros de los listeners y de las conexiones del sistema.
Almacén de Datos	mlt.alarmas_activas	STACK, TIPO, FECHA, ESTADO	Información de las alarmas activas en el sistema.

#### IV.2.4. DFD conceptual de segundo nivel. SIAC, SIAB.



## MINIESPECIFICACIONES

Tipo	Nombre	Atributos	Descripción
Proceso	Mostrar Datos STACK	-	Proceso mediante el cual se extraen los datos del stack especificado.
Proceso	Mostrar Datos Historicos STACK	-	Proceso a petición del usuario mediante el cual se extraen los datos históricos del stack especificado.
Proceso	Mostrar Datos INTERFACES	-	Proceso mediante el cual se extraen los datos de los interfaces especificados.
Proceso	Mostrar Datos Historicos INTERFACES	-	Proceso mediante el cual se extraen los datos históricos del interface especificado.
Proceso	Mostrar Datos LISTENERS	-	Proceso mediante el cual se extraen los datos de los listeners del stack especificado.
Proceso	Mostrar Datos Historicos LISTENERS	-	Proceso mediante el cual se extraen los datos históricos del listener especificado.
Proceso	Mostrar Datos VIPA	-	Proceso mediante el cual se extraen los datos de los interfaces virtuales.
Proceso	Mostrar Datos SERMEPA	-	Proceso mediante el cual se extraen los datos de las conexiones.
Proceso	Presentación	-	Proceso mediante el cual se muestran los datos extraídos.
Flujo de Datos	Stack	-	Nombre del stack del cual queremos ejecutar el proceso.
Flujo de Datos	Interfaces	-	Nombre del interface del cual queremos ejecutar el proceso.
Flujo de Datos	Vipas	-	Nombre de la VIPA sobre la que queremos ejecutar el proceso.
Flujo de Datos	Conexiones	-	Nombre de la conexión sobre la que queremos ejecutar el proceso.
Flujo de Datos	Solicitud Datos STACK	-	Solicitud automatizada por la que se pide la extracción de los datos del stack especificado.
Flujo de Datos	Solicitud Datos Históricos STACK	-	Solicitud a petición del usuario por la que se pide la extracción de los datos históricos del stack especificado.
Flujo de Datos	Solicitud Datos INTERFACES	-	Solicitud automatizada por la que se pide la extracción de los datos de los interfaces especificado.
Flujo de Datos	Solicitud Datos Históricos INTERFACES	-	Solicitud a petición del usuario por la que se pide la extracción de los datos históricos del interface especificado.
Flujo de Datos	Solicitud Datos LISTENERS	-	Solicitud automatizada por la que se pide la extracción de los listeners del stack especificado.

Flujo de Datos	Solicitud Datos Históricos LISTENERS	-	Solicitud a petición del usuario por la que se pide la extracción de los datos históricos del listener especificado.
Flujo de Datos	Solicitud Datos VIPA	-	Solicitud automatizada por la que se pide la extracción de los interfaces virtuales especificados.
Flujo de Datos	Solicitud Datos SERMEPA	-	Solicitud automatizada por la que se pide la extracción de las conexiones especificadas.
Almacén de Datos	mlt.descripcion	STACK, NOMBRE, PUERTO, DESCRIPCION, HORA_D, HORA_H, DIAS_SEM	Información sobre los listeners que se desea monitorizar en el sistema.
Almacén de Datos	mlt.stack_tcp	STACK, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.stack_tcp_hst	STACK, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.stack_tcp_sum_hora	STACK, CAMPO, FECHA, VALOR	Información sumariada por horas de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.stack_tcp_sum_dia	STACK, CAMPO, FECHA, VALOR	Información sumariada por día de los distintos parámetros de los stacks del sistema.
Almacén de Datos	mlt.interfaces	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_hst	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_sum_hora	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información sumariada por horas de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_sum_dia	STACK, DIR_IP, CAMPO, FECHA, VALOR	Información sumariada por día de los distintos parámetros de los interfaces del sistema.
Almacén de Datos	mlt.interfaces_vipa	STACK, PUERTO_IP, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los interfaces virtuales del sistema.
Almacén de Datos	mlt.interfaces_vipa_hst	STACK, PUERTO_IP, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los interfaces virtuales del sistema.
Almacén de Datos	mlt.sockets	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información de los distintos parámetros de los listeners y de las conexiones del sistema.
Almacén de Datos	mlt.sockets_hst	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información histórica de los distintos parámetros de los listeners y de las conexiones del sistema.

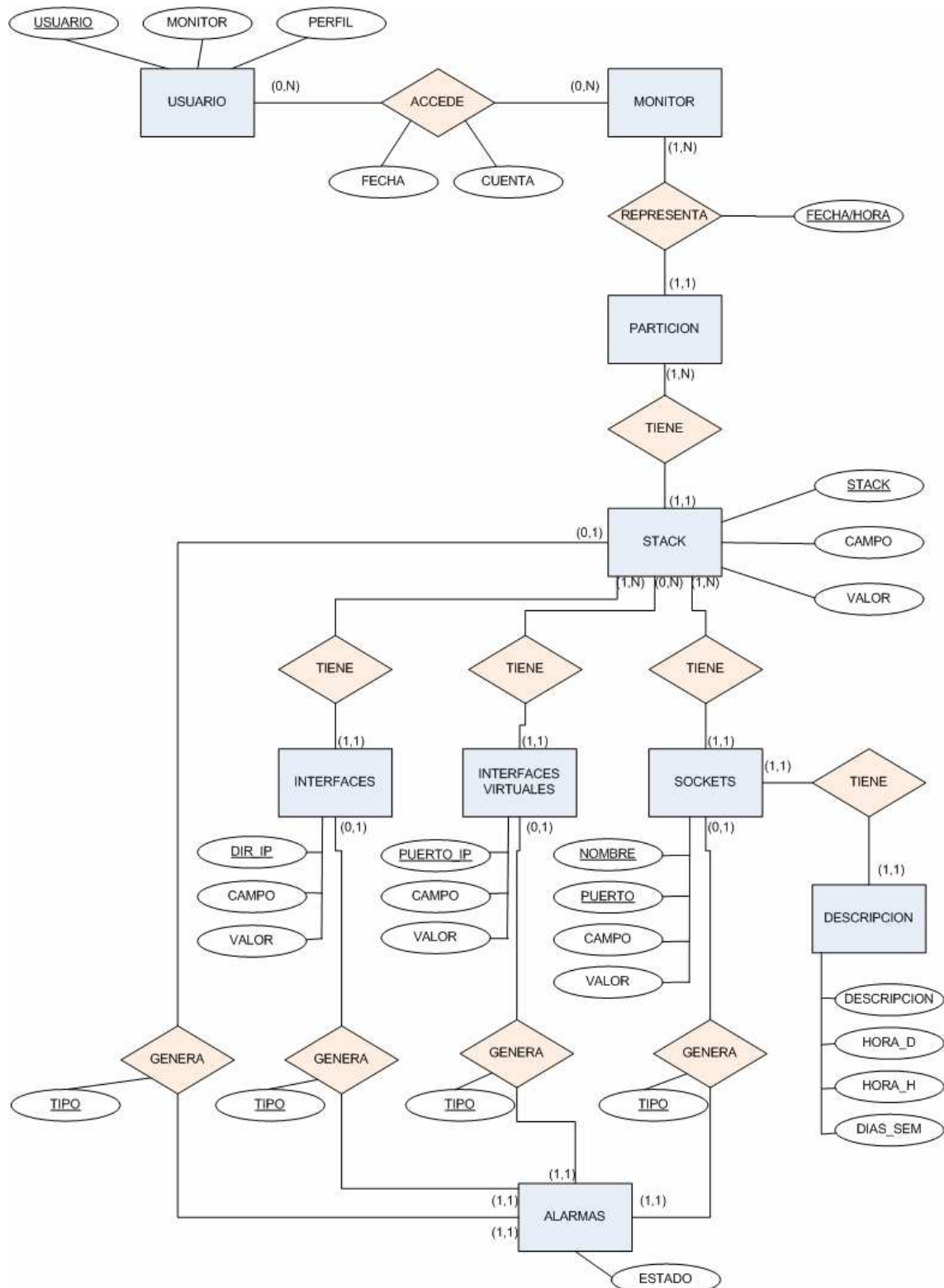
Almacén de Datos	mlt.sockets_sum_hora	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información sumariada por horas de los distintos parámetros de los listeners y de las conexiones del sistema.
Almacén de Datos	mlt.sockets_sum_dia	STACK, NOMBRE, PUERTO, CAMPO, FECHA, VALOR	Información sumariada por dia de los distintos parámetros de los listeners y de las conexiones del sistema.
Almacén de Datos	mlt.alarmas_activas	STACK, TIPO, FECHA, ESTADO	Información de las alarmas activas en el sistema.



### IV.3. Modelo conceptual de datos.

#### IV.3.1. Modelo entidad-relacion.

A continuación se detalla el diagrama entidad-relación de la aplicación después de haberse realizado la normalización de los datos.



## IV.4. Modelo físico.

### IV.4.1. Definiciones de la Base de Datos.

A continuación procedemos a describir cada una de las tablas que tenemos definidas en nuestra Base de Datos Oracle.

## Tabla MONI.USUARIOS

### Contenido

Esta tabla contiene los usuarios que tienen acceso al sistema de monitorización así como los monitores a los que pueden acceder y su nivel de acceso.

### Campos

CAMPO	DESCRIPCION	VALORES
USUARIO	Contiene el nombre del usuario que puede acceder a los monitores.	Alfanumérico
MONITOR	Nombre del/los monitores a los que tiene acceso.	Alfanumérico
PERFIL	Perfil de acceso.	Numérico

### Sentencias SQL

```
-- Create table
create table MONI.USUARIOS
(
  USUARIO CHAR(8) not null,
  MONITOR CHAR(10) not null,
  PERFIL  NUMBER(1)
)
tablespace DATOS
pctfree 10
initrans 1
maxtrans 255
storage
(
  initial 1M
  next 1M
  minextents 1
  maxextents unlimited
);
-- Create/Recreate indexes
create unique index MONI.USUARIOS01 on MONI.USUARIOS (USUARIO,
```

```

MONITOR)
  tablespace DATOS
  pctfree 10
  initrans 2
  maxtrans 255
  storage
  (
    initial 1M
    next 1M
    minextents 1
    maxextents unlimited
  );
-- Grant/Revoke object privileges
grant select on MONI.USUARIOS to PUBLIC;

```

#### Muestra de registros

USUARIO	MONITOR	PERFIL
ex451219	MLT	0
bc009986	MLE	0
bc020705	ALL	0
bc009986	IRS	0
bc037556	IRS	1
bc020705	MLU	0
bc037598	MLL	2
bc049328	MLG	0

## Tabla MONI.ACCESOS

### Contenido

Esta tabla contiene un log del acceso de los usuarios a los distintos monitores.

### Campos

CAMPO	DESCRIPCION	VALORES
USUARIO	Contiene el nombre del usuario que puede acceder a los monitores.	Alfanumérico
MONITOR	Nombre del/los monitores a los que tiene acceso.	Alfanumérico
FECHA	Fecha/Hora del último acceso.	Fecha
CUENTA	Número de accesos al monitor por el usuario	Numérico

### Sentencias SQL

```
-- Create table
create table MONI.ACCESOS
(
  USUARIO CHAR(8) not null,
  MONITOR CHAR(8) not null,
  FECHA DATE,
  CUENTA NUMBER
)
tablespace DATOS
pctfree 10
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
);
-- Create/Recreate indexes
create unique index MONI.ACCESOS01 on MONI.ACCESOS (USUARIO,
MONITOR)
tablespace DATOS
pctfree 10
initrans 2
maxtrans 255
storage
```

```
(  
  initial 64K  
  next 1M  
  minextents 1  
  maxextents unlimited  
);  
  
-- Grant/Revoke object privileges  
grant select, insert, update, delete on MONI.ACCESOS to PUBLIC;
```

#### Muestra de registros

USUARIO	MONITOR	FECHA	CUENTA
bc030586	MLI	19/07/2010 11:49:00	74
bc146447	MLT	12/08/2010 11:58:00	8
bc030586	MLX	18/08/2010 11:26:00	49
e150473	MLO	17/08/2010 8:02:00	391
bc027762	MLO	19/08/2010 13:00:00	91
bc146447	MLQ	12/08/2010 11:58:00	95
bc038364	MLS	16/08/2010 10:10:00	110
bc030586	MLI	19/07/2010 11:49:00	74

## Tabla MLT.DESCRIPCION

### Contenido

Esta tabla contiene la descripción de todos los listeners que van a ser monitorizados así como el intervalo horario en el que tendrá que estar activa dicha monitorización.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK al que pertenece el listener	EXPL, VTL3, SIAB, SIAC, CEIN
NOMBRE	Nombre de la Started Task del Listener	Alfanumérico
PUERTO	Número de Puerto del listener	Numérico
DESCRIPCION	Descripción del listener	Alfanumérico
HORA_D	Hora de comienzo de la monitorización en formato HHMM	Entre 0 y 9999
HORA_H	Hora de finalización de la monitorización en formato HHMM	Entre 0 y 9999
DIAS_SEM	Días de la semana que está activa la monitorización	Entre 0 y 6 (Donde el 0 es el Domingo y el 6 el Sábado)

### Sentencias SQL

```
-- Create table
create table DESCRIPCION
(
    STACK          CHAR(10) not null,
    NOMBRE         CHAR(15) not null,
    PUERTO         NUMBER(5) not null,
    DESCRIPCION    VARCHAR2(50),
    HORA_D         NUMBER(5),
    HORA_H         NUMBER(5),
    DIAS_SEM       VARCHAR2(50)
)
```

```

tablespace DATOS
  pctfree 10
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
  );
-- Create/Recreate indexes
create unique index DESCRIPCION01 on DESCRIPCION (STACK, NOMBRE,
PUERTO)
  tablespace DATOS
  pctfree 10
  initrans 2
  maxtrans 255
  storage
  (
    initial 1M
    next 1M
    minextents 1
    maxextents unlimited
  )
  compress 2;
-- Grant/Revoke object privileges
grant select on DESCRIPCION to MLG;

```

#### Muestra de registros

STACK	NOMBRE	PUERTO	DESCRIPCION	HORA_D	HORA_H	DIAS_SEM
SIAC	MQ1CCHIN	1420	MQ SIA	0	9999	0,1,2,3,4,5,6
SIAC	TN1CPROC	23	Telnet 3270	0	9999	0,1,2,3,4,5,6
SIAC	TN1CPROC	998	Telnet 3270 SSL	0	9999	0,1,2,3,4,5,6
SIAC	FTPIP1C1	21	FTP	0	9999	0,1,2,3,4,5,6
CEIN	DB1NDIST	446	DB2	0	9999	0,1,2,3,4,5,6
EXPL	TPTHTP	8984	Monitorizacion V50	0	9999	0,1,2,3,4,5,6
EXPL	BETA97ME	8798	Gestion Documental (BP)	0	9999	0,1,2,3,4,5,6
EXPL	TPESNP	9000	SNCE	0	9999	0,1,2,3,4,5,6

## Tabla MLT.STACK\_TCP

### Contenido

Esta tabla contiene la información de los STACK's TCP/IP de cada una de las particiones.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK cuyo campo se referencia	EXPL, VTL3, SIAB, SIAC, CEIN
CAMPO	Nombre del campo	xsockets, tcpActiveOpens, tcpPassiveOpens, tcpAttemptFails, tcpEstabResets, tcpOutRsts, tcpInSegs, tcpOutSegs, tcpRetransSegs, tcpInErrs, tcpCurrEstab
FECHA	Fecha/Hora del dato	Fecha/Hora
VALOR	Valor del campo	Numérico

### Sentencias SQL

```
-- Create table
create table STACK_TCP
(
  STACK CHAR(10) not null,
  CAMPO VARCHAR2(30) not null,
  FECHA DATE not null,
  VALOR NUMBER
)
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 128M
  next 1M
  minextents 1
  maxextents unlimited
)
```



```

);
-- Create/Recreate indexes
create unique index STACK_TCP01 on STACK_TCP (STACK, CAMPO,
FECHA)
  tablespace DATOS
  pctfree 10
  initrans 2
  maxtrans 255
  storage
  (
    initial 112M
    next 1M
    minextents 1
    maxextents unlimited
  )
  compress 2;
-- Grant/Revoke object privileges
grant select on STACK_TCP to MLG;

```

#### Muestra de registros

STACK	CAMPO	FECHA	VALOR
EXPL	tcpPassiveOpens	07/08/2010 6:33:00	12954273
EXPL	tcpActiveOpens	07/08/2010 6:33:00	1280659
EXPL	xsockets	07/08/2010 6:33:00	139
EXPL	tcpAttemptFails	07/08/2010 6:33:00	8771
SIAC	tcpCurrEstab	07/08/2010 6:29:00	25
SIAC	tcpInSegs	07/08/2010 6:29:00	130334275
SIAC	tcpOutSegs	07/08/2010 6:29:00	117374403
SIAC	tcpRetransSegs	07/08/2010 6:29:00	18650
SIAC	tcpInErrs	07/08/2010 6:29:00	431
SIAC	tcpOutRsts	07/08/2010 6:29:00	5623
SIAC	tcpEstabResets	07/08/2010 6:29:00	509
SIAC	tcpAttemptFails	07/08/2010 6:29:00	680
SIAC	tcpPassiveOpens	07/08/2010 6:29:00	6489354
SIAC	xsockets	07/08/2010 6:29:00	40
SIAC	tcpActiveOpens	07/08/2010 6:29:00	120771

## Tabla MLT.STACK\_TCP\_HST

### Contenido

Esta tabla contiene la información histórica de los STACK's TCP/IP de cada una de las particiones. Con el fin de optimizar su rendimiento hemos procedido a particionarla.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK cuyo campo se referencia	EXPL, VTL3, SIAB, SIAC, CEIN
CAMPO	Nombre del campo	xsockets, tcpActiveOpens, tcpPassiveOpens, tcpAttemptFails, tcpEstabResets, tcpOutRsts, tcpInSegs, tcpOutSegs, tcpRetransSegs, tcpInErrs, tcpCurrEstab
FECHA	Fecha/Hora del dato	Fecha/Hora
VALOR	Valor del campo	Numérico

### Sentencias SQL

```
-- Create table
create table STACK_TCP_HST
(
  STACK CHAR(10) not null,
  CAMPO VARCHAR2(30) not null,
  FECHA DATE not null,
  VALOR NUMBER
)
partition by range (FECHA)
(
  partition PART_01 values less than (TO_DATE(' 2007-04-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
```

```
storage
(
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
),
partition SYS_P352 values less than (TO_DATE(' 2008-11-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P351 values less than (TO_DATE(' 2008-12-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P350 values less than (TO_DATE(' 2009-01-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P3721 values less than (TO_DATE(' 2009-02-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P353 values less than (TO_DATE(' 2009-03-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
```

```

    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P354 values less than (TO_DATE(' 2009-04-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P355 values less than (TO_DATE(' 2009-05-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P356 values less than (TO_DATE(' 2009-06-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P357 values less than (TO_DATE(' 2009-07-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1

```

```
        maxextents unlimited
    ),
    partition SYS_P358 values less than (TO_DATE(' 2009-08-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P359 values less than (TO_DATE(' 2009-09-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P360 values less than (TO_DATE(' 2009-10-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P361 values less than (TO_DATE(' 2009-11-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P362 values less than (TO_DATE(' 2009-12-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
```

```
(
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
),
partition SYS_P3222 values less than (TO_DATE(' 2010-01-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P3921 values less than (TO_DATE(' 2010-02-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P4062 values less than (TO_DATE(' 2010-03-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P4562 values less than (TO_DATE(' 2010-04-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P4742 values less than (TO_DATE(' 2010-05-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
```

```
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
),
partition SYS_P4921 values less than (TO_DATE(' 2010-06-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
),
partition SYS_P5141 values less than (TO_DATE(' 2010-07-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
),
partition SYS_P5361 values less than (TO_DATE(' 2010-08-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
),
partition SYS_P5581 values less than (TO_DATE(' 2010-09-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
)
```

```

)
)
;
-- Create/Recreate indexes
create unique index STACK_TCP_HST01 on STACK_TCP_HST (FECHA, STACK,
CAMPO);
-- Grant/Revoke object privileges
grant select on STACK_TCP_HST to MLG;

```

### Muestra de registros

STACK	CAMPO	FECHA	VALOR
EXPL	tcpPassiveOpens	07/08/2010 6:33:00	12954273
EXPL	tcpActiveOpens	07/08/2010 6:33:00	1280659
EXPL	xsockets	07/08/2010 6:33:00	139
EXPL	tcpAttemptFails	07/08/2010 6:33:00	8771
SIAC	tcpCurrEstab	07/08/2010 6:29:00	25
SIAC	tcpInSegs	07/08/2010 6:29:00	130334275
SIAC	tcpOutSegs	07/08/2010 6:29:00	117374403
SIAC	tcpRetransSegs	07/08/2010 6:29:00	18650
SIAC	tcpInErrs	07/08/2010 6:29:00	431
SIAC	tcpOutRsts	07/08/2010 6:29:00	5623
SIAC	tcpEstabResets	07/08/2010 6:29:00	509
SIAC	tcpAttemptFails	07/08/2010 6:29:00	680
SIAC	tcpPassiveOpens	07/08/2010 6:29:00	6489354
SIAC	xsockets	07/08/2010 6:29:00	40
SIAC	tcpActiveOpens	07/08/2010 6:29:00	120771



## Tabla MLT.STACK\_TCP\_SUM\_HORA

### Contenido

Esta tabla contiene la información de los STACK's TCP/IP de cada una de las particiones sumariada por horas.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK cuyo campo se referencia	EXPL, VTL3, SIAB, SIAC, CEIN
CAMPO	Nombre del campo	xsockets, tcpActiveOpens, tcpPassiveOpens, tcpAttemptFails, tcpEstabResets, tcpOutRsts, tcpInSegs, tcpOutSegs, tcpRetransSegs, tcpInErrs, tcpCurrEstab
FECHA	Fecha/Hora del dato	Fecha/Hora
VALOR	Valor del campo	Numérico

### Sentencias SQL

```
-- Create table
create table STACK_TCP_SUM_HORA
(
  STACK CHAR(10) not null,
  CAMPO VARCHAR2(30) not null,
  FECHA DATE not null,
  VALOR NUMBER
)
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 8M
  next 1M
```

```

    minextents 1
    maxextents unlimited
);
-- Create/Recreate indexes
create unique index STACK_TCP_SUM_HORA01 on STACK_TCP_SUM_HORA
(FECHA, STACK, CAMPO)
tablespace DATOS
pctfree 10
initrans 2
maxtrans 255
storage
(
    initial 11M
    next 1M
    minextents 1
    maxextents unlimited
);
-- Grant/Revoke object privileges
grant select on STACK_TCP_SUM_HORA to MLG;

```

#### Muestra de registros

STACK	CAMPO	FECHA	VALOR
SIAC	tcpPassiveOpens	30/06/2010 23:00:00	3193
SIAC	tcpCurrEstab	30/06/2010 23:00:00	28
SIAC	tcpActiveOpens	30/06/2010 23:00:00	1120
EXPL	tcpPassiveOpens	30/06/2010 23:00:00	9283
EXPL	tcpCurrEstab	30/06/2010 23:00:00	167
EXPL	tcpActiveOpens	30/06/2010 23:00:00	154
SIAC	tcpPassiveOpens	30/06/2010 22:00:00	3510
SIAC	tcpCurrEstab	30/06/2010 22:00:00	27
SIAC	tcpActiveOpens	30/06/2010 22:00:00	60
EXPL	tcpPassiveOpens	30/06/2010 22:00:00	8236
EXPL	tcpCurrEstab	30/06/2010 22:00:00	221
EXPL	tcpActiveOpens	30/06/2010 22:00:00	166
SIAC	tcpPassiveOpens	30/06/2010 21:00:00	6600
SIAC	tcpCurrEstab	30/06/2010 21:00:00	28
SIAC	tcpActiveOpens	30/06/2010 21:00:00	60

## Tabla MLT.STACK\_TCP\_SUM\_DIA

### Contenido

Esta tabla contiene la información de los STACK's TCP/IP de cada una de las particiones sumariada por días.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK cuyo campo se referencia	EXPL, VTL3, SIAB, SIAC, CEIN
CAMPO	Nombre del campo	xsockets, tcpActiveOpens, tcpPassiveOpens, tcpAttemptFails, tcpEstabResets, tcpOutRsts, tcpInSegs, tcpOutSegs, tcpRetransSegs, tcpInErrs, tcpCurrEstab
FECHA	Fecha del dato	Fecha
VALOR	Valor del campo	Numérico

### Sentencias SQL

```
-- Create table
create table STACK_TCP_SUM_DIA
(
  STACK CHAR(10) not null,
  CAMPO VARCHAR2(30) not null,
  FECHA DATE not null,
  VALOR NUMBER
)
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 1M
  next 1M
```

```

    minextents 1
    maxextents unlimited
);
-- Create/Recreate indexes
create unique index STACK_TCP_SUM_DIA01 on STACK_TCP_SUM_DIA
(FECHA, STACK, CAMPO)
tablespace DATOS
pctfree 10
initrans 2
maxtrans 255
storage
(
    initial 1M
    next 1M
    minextents 1
    maxextents unlimited
)
compress 2;
-- Grant/Revoke object privileges
grant select on STACK_TCP_SUM_DIA to MLG;

```

#### Muestra de registros

STACK	CAMPO	FECHA	VALOR
SIAC	tcpPassiveOpens	30/06/2010	174425
SIAC	tcpCurrEstab	30/06/2010	35
SIAC	tcpActiveOpens	30/06/2010	2538
EXPL	tcpPassiveOpens	30/06/2010	328763
EXPL	tcpCurrEstab	30/06/2010	5121
EXPL	tcpActiveOpens	30/06/2010	34151
SIAC	tcpPassiveOpens	29/06/2010	150797
SIAC	tcpCurrEstab	29/06/2010	34
SIAC	tcpActiveOpens	29/06/2010	2531
EXPL	tcpPassiveOpens	29/06/2010	290810
EXPL	tcpCurrEstab	29/06/2010	4909
EXPL	tcpActiveOpens	29/06/2010	33989
SIAC	tcpPassiveOpens	28/06/2010	160354
SIAC	tcpCurrEstab	28/06/2010	39
SIAC	tcpActiveOpens	28/06/2010	2459

## Tabla MLT.INTERFACES\_VIPA

### Contenido

Esta tabla contiene la información de los interfaces virtuales (VIPA) de las particiones SIAC y SIAB que son las únicas que tienen definidos interfaces de ese tipo.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK cuyo campo se referencia	SIAB, SIAC
PUERTO_IP	Contiene las direcciones IP y los PUERTOS que se van a monitorizar.	Alfanumérico
CAMPO	Nombre del campo	ConnEstado, PortTotalConn, DVIPASstatus, Inbytes
FECHA	Fecha/Hora del dato	Fecha/Hora
VALOR	Valor del campo	Alfanumérico

### Sentencias SQL

```
-- Create table
create table INTERFACES_VIPA
(
    STACK          CHAR(10) not null,
    PUERTO_IP      VARCHAR2(50) not null,
    CAMPO          VARCHAR2(30) not null,
    FECHA          DATE not null,
    VALOR          VARCHAR2(30)
)
tablespace DATOS
pctfree 10
initrans 1
maxtrans 255
storage
(
    initial 160M
    next 1M
    minextents 1
    maxextents unlimited
);
-- Create/Recreate indexes
create unique index INTERFACES_VIPA01 on INTERFACES_VIPA (STACK,
```

```

PUERTO_IP, CAMPO, FECHA)
  tablespace DATOS
  pctfree 10
  initrans 2
  maxtrans 255
  storage
  (
    initial 344M
    next 1M
    minextents 1
    maxextents unlimited
  );
-- Grant/Revoke object privileges
grant select on INTERFACES_VIPA to MLG;

```

### Muestra de registros

STACK	PUERTO_IP	CAMPO	FECHA	VALOR
SIAC	10.100.0.5.13934.10.99.5.32.32077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.21496.10.99.5.32.22077	inbytes	07/08/2010 7:23:00	0
SIAC	10.100.0.5.13937.10.99.5.32.12077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13938.10.99.5.32.42077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13939.10.99.5.32.22077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13940.10.99.5.32.12077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13941.10.99.5.32.42077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13942.10.99.5.32.32077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13943.10.99.5.32.52077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.21496.10.99.5.32.22077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13934.10.99.5.32.32077	inbytes	07/08/2010 7:23:00	0
SIAC	10.100.0.5.13936.10.99.5.32.52077	inbytes	07/08/2010 7:23:00	0
SIAC	10.100.0.5.13937.10.99.5.32.12077	inbytes	07/08/2010 7:23:00	0
SIAC	10.100.0.5.13938.10.99.5.32.42077	inbytes	07/08/2010 7:23:00	0

## Tabla MLT.INTERFACES\_VIPA\_HST

### Contenido

Esta tabla contiene la información histórica de los interfaces virtuales (VIPA) de las particiones SIAC y SIAB que son las únicas que tienen definidos interfaces de ese tipo. Con el fin de optimizar su rendimiento hemos procedido a particionarla.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK cuyo campo se referencia	SIAB, SIAC
PUERTO_IP	Contiene las direcciones IP y los PUERTOS que se van a monitorizar.	Alfanumérico
CAMPO	Nombre del campo	ConnEstado, PortTotalConn, DVIPASStatus, Inbytes
FECHA	Fecha/Hora del dato	Fecha/Hora
VALOR	Valor del campo	Alfanumérico

### Sentencias SQL

```
-- Create table
create table INTERFACES_VIPA_HST
(
    STACK          CHAR(10) not null,
    PUERTO_IP      VARCHAR2(50) not null,
    CAMPO          VARCHAR2(30) not null,
    FECHA          DATE not null,
    VALOR          VARCHAR2(30)
)
partition by range (FECHA)
(
    partition PART_01 values less than (TO_DATE(' 2007-04-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
        )

```

```
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P363 values less than (TO_DATE(' 2009-05-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P364 values less than (TO_DATE(' 2009-06-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P365 values less than (TO_DATE(' 2009-07-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P366 values less than (TO_DATE(' 2009-08-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P367 values less than (TO_DATE(' 2009-09-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
```



```
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
),
partition SYS_P368 values less than (TO_DATE(' 2009-10-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  tablespace DATOS
  pctfree 0
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
  ),
partition SYS_P369 values less than (TO_DATE(' 2009-11-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  tablespace DATOS
  pctfree 0
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
  ),
partition SYS_P370 values less than (TO_DATE(' 2009-12-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  tablespace DATOS
  pctfree 0
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
  ),
partition SYS_P3201 values less than (TO_DATE(' 2010-01-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  tablespace DATOS
  pctfree 0
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
  ),
partition SYS_P3941 values less than (TO_DATE(' 2010-02-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
```

```
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
),
partition SYS_P4064 values less than (TO_DATE(' 2010-03-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
),
partition SYS_P4564 values less than (TO_DATE(' 2010-04-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
),
partition SYS_P4744 values less than (TO_DATE(' 2010-05-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
  maxextents unlimited
),
partition SYS_P4923 values less than (TO_DATE(' 2010-06-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
  initial 64K
  next 1M
  minextents 1
```

```
        maxextents unlimited
    ),
    partition SYS_P5143 values less than (TO_DATE(' 2010-07-01
00:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P5363 values less than (TO_DATE(' 2010-08-01
00:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P5583 values less than (TO_DATE(' 2010-09-01
00:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    )
)
;

-- Create/Recreate indexes
create unique index INTERFACES_VIPA_HST01 on INTERFACES_VIPA_HST
(STACK, PUERTO_IP, CAMPO, FECHA);
-- Grant/Revoke object privileges
grant select on INTERFACES_VIPA_HST to MLG;
```

### Muestra de registros

STACK	PUERTO_IP	CAMPO	FECHA	VALOR
SIAC	10.100.0.5.13934.10.99.5.32.32077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.21496.10.99.5.32.22077	inbytes	07/08/2010 7:23:00	0
SIAC	10.100.0.5.13937.10.99.5.32.12077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13938.10.99.5.32.42077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13939.10.99.5.32.22077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13940.10.99.5.32.12077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13941.10.99.5.32.42077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13942.10.99.5.32.32077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13943.10.99.5.32.52077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.21496.10.99.5.32.22077	ConnEstado	07/08/2010 7:23:00	established
SIAC	10.100.0.5.13934.10.99.5.32.32077	inbytes	07/08/2010 7:23:00	0
SIAC	10.100.0.5.13936.10.99.5.32.52077	inbytes	07/08/2010 7:23:00	0
SIAC	10.100.0.5.13937.10.99.5.32.12077	inbytes	07/08/2010 7:23:00	0
SIAC	10.100.0.5.13938.10.99.5.32.42077	inbytes	07/08/2010 7:23:00	0
SIAC	10.100.0.5.13939.10.99.5.32.22077	inbytes	07/08/2010 7:23:00	389781629
SIAC	10.100.0.5.13940.10.99.5.32.12077	inbytes	07/08/2010 7:23:00	36523272

## Tabla MLT.INTERFACES

### Contenido

Esta tabla contiene la información de los interfaces físicos de las distintas particiones.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK cuyo campo se referencia	EXPL, VTL3, SIAB, SIAC, CEIN
DIR_IP	Contiene las direcciones IP de los interfaces físicos	Alfanumérico
CAMPO	Nombre del campo	ifOutErrors, ifInErrors, ifOutOctets, ifInOctets, ifOperstatus, ifInDiscards
FECHA	Fecha/Hora del dato	Fecha/Hora
VALOR	Valor del campo	Alfanumérico

### Sentencias SQL

```
-- Create table
create table INTERFACES
(
    STACK    CHAR(10) not null,
    DIR_IP   VARCHAR2(30) not null,
    CAMPO    VARCHAR2(30) not null,
    FECHA    DATE not null,
    VALOR    VARCHAR2(30)
)
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
    initial 24M
    next 1M
    minextents 1
    maxextents unlimited
);
-- Create/Recreate indexes
create unique index INTERFACES01 on INTERFACES (STACK, DIR_IP,
CAMPO, FECHA)
tablespace DATOS
```

```
pctfree 10
initrans 2
maxtrans 255
storage
(
  initial 15M
  next 1M
  minextents 1
  maxextents unlimited
)
compress 3;
-- Grant/Revoke object privileges
grant select on INTERFACES to MLG;
```

### Muestra de registros

STACK	DIR_IP	CAMPO	FECHA	VALOR
SIAC	VI1.10.100.0.5	ifOperStatus	09/08/2010 13:27:00	up
SIAC	IF2.10.254.32.80	ifOutErrors	09/08/2010 13:27:00	0
SIAC	IF2.10.254.32.80	ifOutDiscards	09/08/2010 13:27:00	0
SIAC	IF2.10.254.32.80	ifOutOctets	09/08/2010 13:27:00	158282382
SIAC	IF2.10.254.32.80	ifInErrors	09/08/2010 13:27:00	88925
SIAC	IF2.10.254.32.80	ifInDiscards	09/08/2010 13:27:00	0
SIAC	IF2.10.254.32.80	ifInOctets	09/08/2010 13:27:00	3435032247
SIAC	IF2.10.254.32.80	ifOperStatus	09/08/2010 13:27:00	up
SIAC	IF1.10.254.32.42	ifOutErrors	09/08/2010 13:27:00	0
SIAC	IF1.10.254.32.42	ifOutDiscards	09/08/2010 13:27:00	0
SIAC	IF1.10.254.32.42	ifOutOctets	09/08/2010 13:27:00	249067966
SIAC	IF1.10.254.32.42	ifInErrors	09/08/2010 13:27:00	88949
SIAC	IF1.10.254.32.42	ifInDiscards	09/08/2010 13:27:00	0
SIAC	IF1.10.254.32.42	ifInOctets	09/08/2010 13:27:00	3453503126
SIAC	IF1.10.254.32.42	ifOperStatus	09/08/2010 13:27:00	up
SIAC	VI1.10.100.0.5	ifOutErrors	09/08/2010 13:27:00	0
SIAC	VI1.10.100.0.5	ifOutDiscards	09/08/2010 13:27:00	0
SIAC	VI1.10.100.0.5	ifOutOctets	09/08/2010 13:27:00	0
SIAC	VI1.10.100.0.5	ifInErrors	09/08/2010 13:27:00	0

## Tabla MLT.INTERFACES\_HST

### Contenido

Esta tabla contiene la información histórica de los interfaces físicos de las distintas particiones. Con el fin de optimizar su rendimiento hemos procedido a particionarla.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK cuyo campo se referencia	EXPL, VTL3, SIAB, SIAC, CEIN
DIR_IP	Contiene las direcciones IP de los interfaces físicos	Alfanumérico
CAMPO	Nombre del campo	ifOutErrors, ifInErrors, ifOutOctets, ifInOctets, ifOperstatus, ifInDiscards
FECHA	Fecha/Hora del dato	Fecha/Hora
VALOR	Valor del campo	Alfanumérico

### Sentencias SQL

```
-- Create table
create table INTERFACES_HST
(
  STACK CHAR(10) not null,
  DIR_IP VARCHAR2(30) not null,
  CAMPO VARCHAR2(30) not null,
  FECHA DATE not null,
  VALOR VARCHAR2(30)
)
partition by range (FECHA)
(
  partition PART_01 values less than (TO_DATE(' 2007-04-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
      initial 64K
      next 1M
      minextents 1
      maxextents unlimited
    ),

```

```
partition SYS_P326 values less than (TO_DATE(' 2008-11-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  tablespace DATOS
  pctfree 0
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
  ),
partition SYS_P325 values less than (TO_DATE(' 2008-12-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  tablespace DATOS
  pctfree 0
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
  ),
partition SYS_P324 values less than (TO_DATE(' 2009-01-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  tablespace DATOS
  pctfree 0
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
  ),
partition SYS_P323 values less than (TO_DATE(' 2009-02-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  tablespace DATOS
  pctfree 0
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
  ),
partition SYS_P328 values less than (TO_DATE(' 2009-03-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
  tablespace DATOS
  pctfree 0
  initrans 1
  maxtrans 255
  storage
  (
    initial 64K
```



```
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P329 values less than (TO_DATE(' 2009-04-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P330 values less than (TO_DATE(' 2009-05-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P331 values less than (TO_DATE(' 2009-06-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P332 values less than (TO_DATE(' 2009-07-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
        maxtrans 255
        storage
        (
            initial 64K
            next 1M
            minextents 1
            maxextents unlimited
        ),
    partition SYS_P333 values less than (TO_DATE(' 2009-08-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
        tablespace DATOS
        pctfree 0
        initrans 1
```

```
maxtrans 255
storage
(
    initial 64K
    next 1M
    minextents 1
    maxextents unlimited
),
partition SYS_P334 values less than (TO_DATE(' 2009-09-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P3741 values less than (TO_DATE(' 2009-10-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P322 values less than (TO_DATE(' 2009-11-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P327 values less than (TO_DATE(' 2009-12-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
partition SYS_P3181 values less than (TO_DATE(' 2010-01-01
```

```
00:00:00', 'SYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P3922 values less than (TO_DATE(' 2010-02-01
00:00:00', 'SYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P4063 values less than (TO_DATE(' 2010-03-01
00:00:00', 'SYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P4563 values less than (TO_DATE(' 2010-04-01
00:00:00', 'SYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P4743 values less than (TO_DATE(' 2010-05-01
00:00:00', 'SYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
```

```
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P4922 values less than (TO_DATE(' 2010-06-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P5142 values less than (TO_DATE(' 2010-07-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P5362 values less than (TO_DATE(' 2010-08-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P5582 values less than (TO_DATE(' 2010-09-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    )
)
;

-- Create/Recreate indexes
create unique index INTERFACES_HST01 on INTERFACES_HST (FECHA, STACK,
DIR_IP, CAMPO);
-- Grant/Revoke object privileges
```

```
grant select on INTERFACES_HST to MLG;
```

### Muestra de registros

STACK	DIR_IP	CAMPO	FECHA	VALOR
SIAC	VI1.10.100.0.5	ifOperStatus	09/08/2010 13:27:00	up
SIAC	IF2.10.254.32.80	ifOutErrors	09/08/2010 13:27:00	0
SIAC	IF2.10.254.32.80	ifOutDiscards	09/08/2010 13:27:00	0
SIAC	IF2.10.254.32.80	ifOutOctets	09/08/2010 13:27:00	158282382
SIAC	IF2.10.254.32.80	ifInErrors	09/08/2010 13:27:00	88925
SIAC	IF2.10.254.32.80	ifInDiscards	09/08/2010 13:27:00	0
SIAC	IF2.10.254.32.80	ifInOctets	09/08/2010 13:27:00	3435032247
SIAC	IF2.10.254.32.80	ifOperStatus	09/08/2010 13:27:00	up
SIAC	IF1.10.254.32.42	ifOutErrors	09/08/2010 13:27:00	0
SIAC	IF1.10.254.32.42	ifOutDiscards	09/08/2010 13:27:00	0
SIAC	IF1.10.254.32.42	ifOutOctets	09/08/2010 13:27:00	249067966
SIAC	IF1.10.254.32.42	ifInErrors	09/08/2010 13:27:00	88949
SIAC	IF1.10.254.32.42	ifInDiscards	09/08/2010 13:27:00	0
SIAC	IF1.10.254.32.42	ifInOctets	09/08/2010 13:27:00	3453503126
SIAC	IF1.10.254.32.42	ifOperStatus	09/08/2010 13:27:00	up
SIAC	VI1.10.100.0.5	ifOutErrors	09/08/2010 13:27:00	0
SIAC	VI1.10.100.0.5	ifOutDiscards	09/08/2010 13:27:00	0
SIAC	VI1.10.100.0.5	ifOutOctets	09/08/2010 13:27:00	0
SIAC	VI1.10.100.0.5	ifInErrors	09/08/2010 13:27:00	0
SIAC	VI1.10.100.0.5	ifInDiscards	09/08/2010 13:27:00	0
SIAC	VI1.10.100.0.5	ifInOctets	09/08/2010 13:27:00	0
SIAC	VI1.10.100.0.5	ifOperStatus	09/08/2010 13:25:00	up
SIAC	VI1.10.100.0.5	ifInErrors	09/08/2010 13:25:00	0
SIAC	VI1.10.100.0.5	ifInDiscards	09/08/2010 13:25:00	0

## Tabla MLT.INTERFACES\_SUM\_HORA

### Contenido

Esta tabla contiene la información de los interfaces físicos de las distintas particiones sumariados por horas.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK cuyo campo se referencia	EXPL, VTL3, SIAB, SIAC, CEIN
DIR_IP	Contiene las direcciones IP de los interfaces físicos	Alfanumérico
CAMPO	Nombre del campo	ifOutErrors, ifInErrors, ifOutOctets, ifInOctets, ifOperstatus, ifInDiscards
FECHA	Fecha/Hora del dato	Fecha/Hora
VALOR	Valor del campo	Alfanumérico

### Sentencias SQL

```
-- Create table
create table INTERFACES_SUM_HORA
(
    STACK    CHAR(10) not null,
    DIR_IP   VARCHAR2(30) not null,
    CAMPO    VARCHAR2(30) not null,
    FECHA    DATE not null,
    VALOR    VARCHAR2(30)
)
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
    initial 15M
    next 1M
    minextents 1
    maxextents unlimited
);
-- Create/Recreate indexes
create unique index INTERFACES_SUM_HORA01 on INTERFACES_SUM_HORA
(FECHA, STACK, DIR_IP, CAMPO)
```

```

tablespace DATOS
pctfree 10
initrans 2
maxtrans 255
storage
(
  initial 20M
  next 1M
  minextents 1
  maxextents unlimited
);
-- Grant/Revoke object privileges
grant select on INTERFACES_SUM_HORA to MLG;

```

### Muestra de registros

STACK	DIR_IP	CAMPO	FECHA	VALOR
SIAC	VI1.10.100.0.5	ifOutOctets	30/06/2010 23:00:00	0
SIAC	VI1.10.100.0.5	ifInOctets	30/06/2010 23:00:00	0
SIAC	IF2.10.254.32.80	ifOutOctets	30/06/2010 23:00:00	76538
SIAC	IF2.10.254.32.80	ifInOctets	30/06/2010 23:00:00	14540119
SIAC	IF1.10.254.32.42	ifOutOctets	30/06/2010 23:00:00	25497625
SIAC	IF1.10.254.32.42	ifInOctets	30/06/2010 23:00:00	7310996
SIAC	VI1.10.100.0.5	ifOutOctets	30/06/2010 22:00:00	0
SIAC	VI1.10.100.0.5	ifInOctets	30/06/2010 22:00:00	0
SIAC	IF2.10.254.32.80	ifOutOctets	30/06/2010 22:00:00	77044
SIAC	IF2.10.254.32.80	ifInOctets	30/06/2010 22:00:00	23630464
SIAC	IF1.10.254.32.42	ifOutOctets	30/06/2010 22:00:00	49010774
SIAC	IF1.10.254.32.42	ifInOctets	30/06/2010 22:00:00	10651915
SIAC	VI1.10.100.0.5	ifOutOctets	30/06/2010 21:00:00	0
SIAC	VI1.10.100.0.5	ifInOctets	30/06/2010 21:00:00	0
SIAC	IF2.10.254.32.80	ifOutOctets	30/06/2010 21:00:00	80619
SIAC	IF2.10.254.32.80	ifInOctets	30/06/2010 21:00:00	32238552
SIAC	IF1.10.254.32.42	ifOutOctets	30/06/2010 21:00:00	80758604
SIAC	IF1.10.254.32.42	ifInOctets	30/06/2010 21:00:00	17049880
SIAC	VI1.10.100.0.5	ifOutOctets	30/06/2010 20:00:00	0

## Tabla MLT.INTERFACES\_SUM\_DIA

### Contenido

Esta tabla contiene la información de los interfaces físicos de las distintas particiones sumariados por días.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK cuyo campo se referencia	EXPL, VTL3, SIAB, SIAC, CEIN
DIR_IP	Contiene las direcciones IP de los interfaces físicos	Alfanumérico
CAMPO	Nombre del campo	ifOutErrors, ifInErrors, ifOutOctets, ifInOctets, ifOperstatus, ifInDiscards
FECHA	Fecha del dato	Fecha
VALOR	Valor del campo	Alfanumérico

### Sentencias SQL

```
-- Create table
create table INTERFACES_SUM_DIA
(
    STACK    CHAR(10) not null,
    DIR_IP   VARCHAR2(30) not null,
    CAMPO    VARCHAR2(30) not null,
    FECHA    DATE not null,
    VALOR    VARCHAR2(30)
)
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
    initial 1M
    next 1M
    minextents 1
    maxextents unlimited
);
-- Create/Recreate indexes
create unique index INTERFACES_SUM_DIA01 on INTERFACES_SUM_DIA
(FECHA, STACK, DIR_IP, CAMPO)
```



```

tablespace DATOS
pctfree 10
initrans 2
maxtrans 255
storage
(
    initial 1M
    next 1M
    minextents 1
    maxextents unlimited
)
compress 3;
-- Grant/Revoke object privileges
grant select on INTERFACES_SUM_DIA to MLG;

```

### Muestra de registros

STACK	DIR_IP	CAMPO	FECHA	VALOR
SIAC	VI1.10.100.0.5	ifOutOctets	30/06/2010	0
SIAC	VI1.10.100.0.5	ifInOctets	30/06/2010	0
SIAC	IF2.10.254.32.80	ifOutOctets	30/06/2010	3020787
SIAC	IF2.10.254.32.80	ifInOctets	30/06/2010	732164573
SIAC	IF1.10.254.32.42	ifOutOctets	30/06/2010	1369761618
SIAC	IF1.10.254.32.42	ifInOctets	30/06/2010	474635047
SIAC	VI1.10.100.0.5	ifOutOctets	29/06/2010	0
SIAC	VI1.10.100.0.5	ifInOctets	29/06/2010	0
SIAC	IF2.10.254.32.80	ifOutOctets	29/06/2010	2978782
SIAC	IF2.10.254.32.80	ifInOctets	29/06/2010	698470978
SIAC	IF1.10.254.32.42	ifOutOctets	29/06/2010	1229032965
SIAC	IF1.10.254.32.42	ifInOctets	29/06/2010	446053901
SIAC	VI1.10.100.0.5	ifOutOctets	28/06/2010	0
SIAC	VI1.10.100.0.5	ifInOctets	28/06/2010	0
SIAC	IF2.10.254.32.80	ifOutOctets	28/06/2010	3126166
SIAC	IF2.10.254.32.80	ifInOctets	28/06/2010	752484627
SIAC	IF1.10.254.32.42	ifOutOctets	28/06/2010	1304392919
SIAC	IF1.10.254.32.42	ifInOctets	28/06/2010	462756109

## Tabla MLT.SOCKETS

### Contenido

Esta tabla contiene la información de los sockets/listeners de las distintas particiones.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK al que pertenece el listener	EXPL, VTL3, SIAB, SIAC, CEIN
NOMBRE	Nombre de la Started Task del Listener/Socket	Alfanumérico
PUERTO	Número de Puerto del listener/socket	Numérico
CAMPO	Nombre del campo	AcceptCount. ExceedBacklog, CurrBacklog, MaxBacklog, Established, FinWait2, TimeWait, CloseWait
FECHA	Fecha del dato	Fecha
VALOR	Valor del campo	Alfanumérico

### Sentencias SQL

```
-- Create table
create table SOCKETS
(
  STACK  CHAR(10) not null,
  NOMBRE CHAR(15) not null,
  PUERTO NUMBER(5) not null,
  CAMPO  VARCHAR2(30) not null,
  FECHA  DATE not null,
  VALOR  NUMBER
)
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
```

```

        initial 256M
        next 1M
        minextents 1
        maxextents unlimited
    );
-- Create/Recreate indexes
create unique index SOCKETS01 on SOCKETS (STACK, NOMBRE, PUERTO,
CAMPO, FECHA)
    tablespace DATOS
    pctfree 10
    initrans 2
    maxtrans 255
    storage
    (
        initial 160M
        next 1M
        minextents 1
        maxextents unlimited
    )
    compress 4;
create index SOCKETS02 on SOCKETS (FECHA)
    tablespace DATOS
    pctfree 10
    initrans 2
    maxtrans 255
    storage
    (
        initial 136M
        next 1M
        minextents 1
        maxextents unlimited
    )
    compress;
-- Grant/Revoke object privileges
grant select on SOCKETS to MLG;

```

#### Muestra de registros

STACK	NOMBRE	PUERTO	CAMPO	FECHA	VALOR
SIAC	TPTS1C	8994	MaxBacklog	09/08/2010 13:31:00	40
SIAC	TPTS1C	8994	CurrBacklog	09/08/2010 13:31:00	0
SIAC	TPTS1C	8994	ExceedBacklog	09/08/2010 13:31:00	0
SIAC	TPTS1C	8994	AcceptCount	09/08/2010 13:31:00	1387
SIAC	TPTS1C	8996	CloseWait	09/08/2010 13:31:00	0
SIAC	TPTS1C	8996	TimeWait	09/08/2010 13:31:00	0
SIAC	TPTS1C	8996	FinWait2	09/08/2010 13:31:00	0
SIAC	TPTS1C	8996	Established	09/08/2010 13:31:00	0

## Tabla MLT.SOCKETS\_HST

### Contenido

Esta tabla contiene la información histórica de los sockets/listeners de las distintas particiones. Con el fin de optimizar su rendimiento hemos procedido a particionarla.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK al que pertenece el listener	EXPL, VTL3, SIAB, SIAC, CEIN
NOMBRE	Nombre de la Started Task del Listener/Socket	Alfanumérico
PUERTO	Número de Puerto del listener/socket	Numérico
CAMPO	Nombre del campo	AcceptCount, ExceedBacklog, CurrBacklog, MaxBacklog, Established, FinWait2, TimeWait, CloseWait
FECHA	Fecha del dato	Fecha
VALOR	Valor del campo	Alfanumérico

### Sentencias SQL

```
-- Create table
create table SOCKETS_HST
(
  STACK CHAR(10) not null,
  NOMBRE CHAR(15) not null,
  PUERTO NUMBER(5) not null,
  CAMPO VARCHAR2(30) not null,
  FECHA DATE not null,
  VALOR NUMBER
)
partition by range (FECHA)
(
  partition PART_01 values less than (TO_DATE(' 2007-04-01 00:00:00',
```

```
'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P3901 values less than (TO_DATE(' 2010-02-01
00:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P4061 values less than (TO_DATE(' 2010-03-01
00:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P4561 values less than (TO_DATE(' 2010-04-01
00:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P4741 values less than (TO_DATE(' 2010-05-01
00:00:00', 'YYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
```

```
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P4901 values less than (TO_DATE(' 2010-06-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P5121 values less than (TO_DATE(' 2010-07-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P5341 values less than (TO_DATE(' 2010-08-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    ),
    partition SYS_P5561 values less than (TO_DATE(' 2010-09-01
00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN'))
    tablespace DATOS
    pctfree 0
    initrans 1
    maxtrans 255
    storage
    (
        initial 64K
        next 1M
        minextents 1
        maxextents unlimited
    )
)
;

-- Create/Recreate indexes
create unique index SOCKETS_HST01 on SOCKETS_HST (FECHA, STACK,
NOMBRE, PUERTO, CAMPO);
-- Grant/Revoke object privileges
```

```
grant select on SOCKETS_HST to MLG;
```

#### Muestra de registros

STACK	NOMBRE	PUERTO	CAMPO	FECHA	VALOR
SIAC	TPTS1C	8994	MaxBacklog	09/08/2010 13:31:00	40
SIAC	TPTS1C	8994	CurrBacklog	09/08/2010 13:31:00	0
SIAC	TPTS1C	8994	ExceedBacklog	09/08/2010 13:31:00	0
SIAC	TPTS1C	8994	AcceptCount	09/08/2010 13:31:00	1387
SIAC	TPTS1C	8996	CloseWait	09/08/2010 13:31:00	0
SIAC	TPTS1C	8996	TimeWait	09/08/2010 13:31:00	0
SIAC	TPTS1C	8996	FinWait2	09/08/2010 13:31:00	0
SIAC	TPTS1C	8996	Established	09/08/2010 13:31:00	0
SIAC	TPTS1C	8996	MaxBacklog	09/08/2010 13:31:00	50
SIAC	TPTS1C	8996	CurrBacklog	09/08/2010 13:31:00	0
SIAC	TPTS1C	8996	ExceedBacklog	09/08/2010 13:31:00	0
SIAC	TPTS1C	8996	AcceptCount	09/08/2010 13:31:00	2
SIAC	TPTS1C	8995	CloseWait	09/08/2010 13:31:00	0
SIAC	TPTS1C	8995	TimeWait	09/08/2010 13:31:00	0
SIAC	TPTS1C	8995	FinWait2	09/08/2010 13:31:00	0

## Tabla MLT.SOCKETS\_SUM\_HORA

### Contenido

Esta tabla contiene la información de los sockets/listeners de las distintas particiones sumariada por horas.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK al que pertenece el listener	EXPL, VTL3, SIAB, SIAC, CEIN
NOMBRE	Nombre de la Started Task del Listener/Socket	Alfanumérico
PUERTO	Número de Puerto del listener/socket	Numérico
CAMPO	Nombre del campo	AcceptCount, ExceedBacklog, CurrBacklog, MaxBacklog, Established, FinWait2, TimeWait, CloseWait
FECHA	Fecha del dato	Fecha
VALOR	Valor del campo	Alfanumérico

### Sentencias SQL

```
-- Create table
create table SOCKETS_SUM_HORA
(
  STACK  CHAR(10) not null,
  NOMBRE CHAR(15) not null,
  PUERTO NUMBER(5) not null,
  CAMPO  VARCHAR2(30) not null,
  FECHA  DATE not null,
  VALOR  NUMBER
)
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
```



```

        initial 215M
        next 1M
        minextents 1
        maxextents unlimited
    );
-- Create/Recreate indexes
create unique index SOCKETS_SUM_HORA01 on SOCKETS_SUM_HORA
(FECHA, STACK, NOMBRE, PUERTO, CAMPO)
    tablespace DATOS
    pctfree 10
    initrans 2
    maxtrans 255
    storage
    (
        initial 313M
        next 1M
        minextents 1
        maxextents unlimited
    );
-- Grant/Revoke object privileges
grant select on SOCKETS_SUM_HORA to MLG;

```

#### Muestra de registros

STACK	NOMBRE	PUERTO	CAMPO	FECHA	VALOR
SIAC	TPTS1C	8996	Established	30/06/2010 23:00:00	0
SIAC	TPTS1C	8996	AcceptCount	30/06/2010 23:00:00	0
SIAC	TPTS1C	8995	Established	30/06/2010 23:00:00	5
SIAC	TPTS1C	8995	AcceptCount	30/06/2010 23:00:00	3163
SIAC	TPTS1C	8994	Established	30/06/2010 23:00:00	0
SIAC	TPTS1C	8994	AcceptCount	30/06/2010 23:00:00	0
SIAC	TN1CPROC	998	Established	30/06/2010 23:00:00	0
SIAC	TN1CPROC	998	AcceptCount	30/06/2010 23:00:00	0
SIAC	TN1CPROC	23	Established	30/06/2010 23:00:00	0
SIAC	TN1CPROC	23	AcceptCount	30/06/2010 23:00:00	0
SIAC	OSNMPC1	1027	Established	30/06/2010 23:00:00	0
SIAC	OSNMPC1	1027	AcceptCount	30/06/2010 23:00:00	0

## Tabla MLT.SOCKETS\_SUM\_DIA

### Contenido

Esta tabla contiene la información de los sockets/listeners de las distintas particiones sumariada por días.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK al que pertenece el listener	EXPL, VTL3, SIAB, SIAC, CEIN
NOMBRE	Nombre de la Started Task del Listener/Socket	Alfanumérico
PUERTO	Número de Puerto del listener/socket	Numérico
CAMPO	Nombre del campo	AcceptCount. ExceedBacklog, CurrBacklog, MaxBacklog, Established, FinWait2, TimeWait, CloseWait
FECHA	Fecha del dato	Fecha
VALOR	Valor del campo	Alfanumérico

### Sentencias SQL

```
-- Create table
create table SOCKETS_SUM_DIA
(
  STACK CHAR(10) not null,
  NOMBRE CHAR(15) not null,
  PUERTO NUMBER(5) not null,
  CAMPO VARCHAR2(30) not null,
  FECHA DATE not null,
  VALOR NUMBER
)
tablespace DATOS
pctfree 0
initrans 1
maxtrans 255
storage
(
```

```

        initial 10M
        next 1M
        minextents 1
        maxextents unlimited
    );
-- Create/Recreate indexes
create unique index SOCKETS_SUM_DIA01 on SOCKETS_SUM_DIA (FECHA,
STACK, NOMBRE, PUERTO, CAMPO)
    tablespace DATOS
    pctfree 10
    initrans 2
    maxtrans 255
    storage
    (
        initial 11M
        next 1M
        minextents 1
        maxextents unlimited
    )
    compress 4;
-- Grant/Revoke object privileges
grant select on SOCKETS_SUM_DIA to MLG;

```

#### Muestra de registros

STACK	NOMBRE	PUERTO	CAMPO	FECHA	VALOR
SIAC	TPTS1C	8996	Established	30/06/2010	0
SIAC	TPTS1C	8996	AcceptCount	30/06/2010	0
SIAC	TPTS1C	8995	Established	30/06/2010	8
SIAC	TPTS1C	8995	AcceptCount	30/06/2010	171846
SIAC	TPTS1C	8994	Established	30/06/2010	3
SIAC	TPTS1C	8994	AcceptCount	30/06/2010	1842
SIAC	TN1CPROC	998	Established	30/06/2010	0
SIAC	TN1CPROC	998	AcceptCount	30/06/2010	0
SIAC	TN1CPROC	23	Established	30/06/2010	3
SIAC	TN1CPROC	23	AcceptCount	30/06/2010	15
SIAC	OSNMPC1	1027	Established	30/06/2010	0

## Tabla MLT.ALARMAS\_ACTIVAS

### Contenido

Esta tabla contiene las alarmas que se han generado y su estado para generar el correspondiente sonido de alarma en el caso de que haya alguna activa.

### Campos

CAMPO	DESCRIPCION	VALORES
STACK	Contiene el nombre del STACK al que pertenece el listener	EXPL, VTL3, SIAB, SIAC, CEIN
TIPO	Contiene el tipo de alarma	STACK, INTFIS, INTVIR, SERMEPA, LISTENER
FECHA	Fecha/Hora de la alarma	Fecha de la alarma
ESTADO	Estado de la alarma	ACT o vacío

### Sentencias SQL

```
-- Create table
create table ALARMAS_ACTIVAS
(
  STACK  CHAR(10) not null,
  TIPO   CHAR(15) not null,
  FECHA  DATE not null,
  ESTADO CHAR(3)
)
tablespace DATOS
pctfree 10
initrans 1
maxtrans 255
storage
(
  initial 1M
  next 1M
  minextents 1
  maxextents unlimited
);
-- Create/Recreate indexes
create unique index ALARMAS_ACTIVAS01 on ALARMAS_ACTIVAS (STACK,
TIPO,FECHA)
tablespace DATOS
pctfree 10
initrans 2
maxtrans 255
storage
```

```
(  
  initial 1M  
  next 1M  
  minextents 1  
  maxextents unlimited  
);  
  
-- Grant/Revoke object privileges  
grant select on ALARMAS_ACTIVAS to MLG;
```

Muestra de registros

STACK	TIPO	FECHA	ESTADO
SIAB	STACK	30/06/2010 23:01:00	
SIAC	CONN	31/07/2010 11:06:00	
SIAB	CONN	30/06/2010 13:33:00	
SIAB	INTVIR	28/12/2009 09:17:00	
EXPL	CONN	22/12/2009 21:09:00	

## IV.4.2. Mantenimiento de la Base de Datos.

A continuación vamos a proceder a describir el mantenimiento que se hace sobre cada una de las tablas con el fin de mantener el rendimiento del monitor así como para cargar las tablas históricas que luego se utilizarán por otros programas que no son objeto de este proyecto.

### IV.4.2.1. Diario

#### mlt\_lanza\_historico.sh

Este script realizado en bash\* es el encargado de realizar la descarga de las tablas que utiliza el monitor a las tablas históricas. Está planificado en el crontab del usuario mlt para que se ejecute todos los días a las 03:00 horas. Con este proceso dejamos las tablas con los datos de los 3 últimos días que son más que suficientes para la monitorización online.

#### \* Bourne-Again Shell

*Bash* está basado en la shell de Unix y es compatible con POSIX. Fue escrito para el proyecto GNU y es el intérprete de comandos por defecto en la mayoría de las distribuciones de Linux. Su nombre es un acrónimo de *Bourne-Again Shell* (otro shell bourne)—haciendo un juego de palabras (“bornagain” significa renacimiento) sobre el *Bourne shell* (sh), que fue uno de los primeros intérpretes importantes de Unix.

```
#!/bin/bash

sqlplus -s mlt/mlt @/www/htdocs/monitores/mlt/sh/sql_hst
status=$?
if [[ ${status} -ne 0 ]] ; then
    echo "Error en historificacion mlt" >> /tmp/mlt.salida
fi
```

#### sql\_hst

```
set echo off verify off feedback off
exec mlt.historifica.sockets
exec mlt.historifica.stack_tcp
exec mlt.historifica.interfaces
exec mlt.historifica.interfaces_vipa
exit
```

#### mlt.historifica.sockets

```
create or replace package body historifica is
w_fecha date;
w_sqlerrm varchar2(150);

procedure sockets is
begin
    w_fecha := sysdate -2;
    insert into sockets_hst
    select * from sockets
    where fecha <= w_fecha;
```

```
delete from sockets
where fecha <= w_fecha;

commit;

exception
when others then
    w_sqlerrm := SUBSTR(SQLERRM, 1, 150);
    raise_application_error (-20001,w_sqlerrm);
end sockets;
```

### mlt.historifica.stack\_tcp

```
procedure stack_tcp is
begin
    w_fecha := sysdate -2;
    insert into stack_tcp_hst
    select * from stack_tcp
    where fecha <= w_fecha;

    delete from stack_tcp
    where fecha <= w_fecha;

    commit;

exception
when others then
    w_sqlerrm := SUBSTR(SQLERRM, 1, 150);
    raise_application_error (-20001,w_sqlerrm);
end stack_tcp;
```

### mlt.historifica.interfaces

```
procedure interfaces is
begin
    w_fecha := sysdate -2;
    insert into interfaces_hst
    select * from interfaces
    where fecha <= w_fecha;

    delete from interfaces
    where fecha <= w_fecha;

    commit;

exception
when others then
    w_sqlerrm := SUBSTR(SQLERRM, 1, 150);
    raise_application_error (-20001,w_sqlerrm);
end interfaces;
```

### mlt.historifica.interfaces\_vipa

```
procedure interfaces_vipa is
begin
    w_fecha := sysdate -2;
    insert into interfaces_vipa_hst
    select * from interfaces_vipa
    where fecha <= w_fecha;

    delete from interfaces_vipa
    where fecha <= w_fecha;

    commit;
```

```
exception
  when others then
    w_sqlerrm := SUBSTR(SQLERRM, 1, 150);
    raise_application_error (-20001,w_sqlerrm);
end interfaces_vipa;

end historifica;
```

#### IV.4.2.2. Mensual

Mensualmente como consecuencia de la elaboración de los informes mensuales de extracción de estadísticas, que no forman parte de este proyecto, se procede a la sumarización de las tablas. Esta sumarización se realiza de forma manual ejecutando los scripts realizados a tal efecto.

##### mlt.sumariza.sockets\_conexiones\_sum\_dia

```
create or replace package body sumariza is
procedure sockets_Conexiones_sum_dia is
begin
insert into sockets_sum_dia
select stack, nombre,puerto,campo, trunc(fecha) fecha,
max(valor)
from sockets_hst
where to_char(fecha,'yyyymm') = to_char(add_months(sysdate,-
1),'yyyymm')
and campo = 'Established'
group by stack, nombre,puerto,campo, trunc(fecha)
order by 1,2,3,4;
commit;
end sockets_Conexiones_sum_dia;
```

##### mlt.sumariza.sockets\_conexiones\_sum\_hora

```
procedure sockets_Conexiones_sum_hora is
begin
insert into sockets_sum_hora
select stack, nombre,puerto,campo, trunc(fecha,'hh24') fecha,
max(valor)
from sockets_hst
where to_char(fecha,'yyyymm') = to_char(add_months(sysdate,-
1),'yyyymm')
and campo = 'Established'
group by stack, nombre,puerto,campo, trunc(fecha,'hh24')
order by 1,2,3,4;
commit;
end sockets_Conexiones_sum_hora;
```

##### mlt.sumariza.sockets\_Accept\_sum\_dia

```
procedure sockets_Accept_sum_dia is
begin
insert into sockets_sum_dia
select stack,nombre, puerto, campo,fecha,
case when (ant <=0)
```



```
        then 0
        else ant
    end
from
(select stack, nombre, puerto, campo, fecha, (valor -
(nvl(lag(valor) over (partition by stack, nombre, puerto, campo
order by fecha), 0))) ant
from (select stack, nombre, puerto, campo, trunc(fecha) fecha,
max(valor) valor
from sockets_hst
where (to_char(fecha, 'yyyymm') between
to_char(add_months(sysdate, -2), 'yyyymm') and
to_char(add_months(sysdate, -1), 'yyyymm'))
and campo = 'AcceptCount'
group by stack, nombre, puerto, campo, trunc(fecha))
order by 1, 2, 3, 4)
where to_char(fecha, 'yyyymm') = to_char(add_months(sysdate, -
1), 'yyyymm');
commit;
end sockets_Accept_sum_dia;
```

### mlt.sumariza.sockets\_Accept\_sum\_hora

```
procedure sockets_Accept_sum_hora is
begin
insert into sockets_sum_hora
select stack, nombre, puerto, campo, fecha,
case when (ant <= 0)
then 0
else ant
end
from
(select stack, nombre, puerto, campo, fecha, (valor -
(nvl(lag(valor) over (partition by stack, nombre, puerto, campo
order by fecha), 0))) ant
from (select stack, nombre, puerto, campo, trunc(fecha, 'hh24')
fecha, max(valor) valor
from sockets_hst
where (to_char(fecha, 'yyyymm') between
to_char(add_months(sysdate, -2), 'yyyymm') and
to_char(add_months(sysdate, -1), 'yyyymm'))
and campo = 'AcceptCount'
group by stack, nombre, puerto, campo, trunc(fecha, 'hh24'))
order by 1, 2, 3, 4)
where to_char(fecha, 'yyyymm') = to_char(add_months(sysdate, -
1), 'yyyymm');
commit;
end sockets_Accept_sum_hora;
```

### mlt.sumariza.stack\_CurrEstab\_sum\_dia

```
procedure stack_CurrEstab_sum_dia is
begin
insert into stack_tcp_sum_dia
select stack, campo, trunc(fecha) fecha, max(valor)
from stack_tcp_hst
where to_char(fecha, 'yyyymm') = to_char(add_months(sysdate, -
1), 'yyyymm')
and campo = 'tcpCurrEstab'
group by stack, campo, trunc(fecha)
```

```
order by 1,2,3;  
commit;  
end stack_CurrEstab_sum_dia;
```

### mlt.sumariza.stack\_CurrEstab\_sum\_hora

```
procedure stack_CurrEstab_sum_hora is  
begin  
insert into stack_tcp_sum_hora  
select stack,campo, trunc(fecha,'hh24') fecha, max(valor)  
from stack_tcp_hst  
where to_char(fecha,'yyyymm') = to_char(add_months(sysdate,-  
1),'yyyymm')  
and campo = 'tcpCurrEstab'  
group by stack, campo, trunc(fecha,'hh24')  
order by 1,2,3;  
commit;  
end stack_CurrEstab_sum_hora;
```

### mlt.sumariza.stack\_AOpens\_sum\_dia

```
procedure stack_AOpens_sum_dia is  
begin  
insert into stack_tcp_sum_dia  
select stack,campo,fecha,  
case when (ant <=0)  
then 0  
else ant  
end  
from  
(SELECT stack,campo, fecha,(valor - (nvl(lag(valor) over  
(partition by stack, campo order by fecha),0))) ant  
from (select stack,campo,trunc(fecha) fecha, max(valor) valor  
from MLT.stack_tcp_hst  
where campo = 'tcpActiveOpens'  
and (to_char(fecha,'yyyymm') between  
to_char(add_months(sysdate,-2),'yyyymm') and  
to_char(add_months(sysdate,-1),'yyyymm'))  
GROUP BY stack,campo,trunc(fecha))  
order by fecha desc)  
where to_char(fecha,'yyyymm')= to_char(add_months(sysdate,-  
1),'yyyymm');  
commit;  
end stack_AOpens_sum_dia;
```

### mlt.sumariza.stack\_AOpens\_sum\_hora

```
procedure stack_AOpens_sum_hora is  
begin  
insert into stack_tcp_sum_hora  
select stack,campo,fecha,  
case when (ant <=0)  
then 0  
else ant  
end  
from  
(SELECT stack,campo, fecha,(valor - (nvl(lag(valor) over  
(partition by stack, campo order by fecha),0))) ant  
from (select stack,campo,trunc(fecha,'hh24') fecha, max(valor)  
valor
```

```
from MLT.stack_tcp_hst
where campo = 'tcpActiveOpens'
and (to_char(fecha, 'yyyymm') between
to_char(add_months(sysdate, -2), 'yyyymm') and
to_char(add_months(sysdate, -1), 'yyyymm'))
GROUP BY stack, campo, trunc(fecha, 'hh24'))
order by fecha desc)
where to_char(fecha, 'yyyymm')= to_char(add_months(sysdate, -
1), 'yyyymm');
commit;
end stack_AOpens_sum_hora;
```

## mlt.sumariza.stack\_POpens\_sum\_dia

```
procedure stack_POpens_sum_dia is
begin
insert into stack_tcp_sum_dia
select stack, campo, fecha,
case when (ant <=0)
then 0
else ant
end
from
(SELECT stack, campo, fecha, (valor - (nvl(lag(valor) over
(partition by stack, campo order by fecha), 0))) ant
from (select stack, campo, trunc(fecha) fecha, max(valor) valor
from MLT.stack_tcp_hst
where campo = 'tcpPassiveOpens'
and (to_char(fecha, 'yyyymm') between
to_char(add_months(sysdate, -2), 'yyyymm') and
to_char(add_months(sysdate, -1), 'yyyymm'))
GROUP BY stack, campo, trunc(fecha))
order by fecha desc)
where to_char(fecha, 'yyyymm')= to_char(add_months(sysdate, -
1), 'yyyymm');
commit;
end stack_POpens_sum_dia;
```

## mlt.sumariza.stack\_POpens\_sum\_hora

```
procedure stack_POpens_sum_hora is
begin
insert into stack_tcp_sum_hora
select stack, campo, fecha,
case when (ant <=0)
then 0
else ant
end
from
(SELECT stack, campo, fecha, (valor - (nvl(lag(valor) over
(partition by stack, campo order by fecha), 0))) ant
from (select stack, campo, trunc(fecha, 'hh24') fecha, max(valor)
valor
from MLT.stack_tcp_hst
where campo = 'tcpPassiveOpens'
and (to_char(fecha, 'yyyymm') between
to_char(add_months(sysdate, -2), 'yyyymm') and
to_char(add_months(sysdate, -1), 'yyyymm'))
GROUP BY stack, campo, trunc(fecha, 'hh24'))
```

```
order by fecha desc)
where to_char(fecha,'yyyymm')= to_char(add_months(sysdate,-
1),'yyyymm');
commit;
end stack_POpens_sum_hora;
```

## mlt.sumariza.interfa\_sum\_dia

```
procedure interfa_sum_dia is
begin
insert into interfaces_sum_dia
select stack, dir_ip,campo, trunc(fecha),sum(diferencia)
from
(select stack, dir_ip,campo, fecha, valor,ant,
case when (to_number(valor) < to_number(ant))
then 0
else to_number(valor-ant)
end diferencia
from
(select stack, dir_ip,campo, fecha,valor ,(nvl(lag(valor) over
(partition by stack, dir_ip,campo order by fecha),0)) ant
from
(select stack, dir_ip,campo, fecha, valor
from interfaces_hst
where (to_char(fecha,'yyyymm') between
to_char(add_months(sysdate,-2),'yyyymm') and
to_char(add_months(sysdate,-1),'yyyymm'))
and campo = 'ifInOctets'
order by stack,dir_ip,campo,fecha asc)
order by stack,dir_ip,campo,fecha asc)
order by stack,dir_ip,campo,fecha asc)
where to_char(fecha,'yyyymm')= to_char(add_months(sysdate,-
1),'yyyymm')
group by stack, dir_ip,campo, trunc(fecha)
order by stack,dir_ip,trunc(fecha) asc;
commit;
insert into interfaces_sum_dia
select stack, dir_ip,campo, trunc(fecha), sum(diferencia)
from
(select stack, dir_ip,campo, fecha, valor,ant,
case when (to_number(valor) < to_number(ant))
then 0
else to_number(valor-ant)
end diferencia
from
(select stack, dir_ip,campo, fecha,valor ,(nvl(lag(valor) over
(partition by stack, dir_ip,campo order by fecha),0)) ant
from
(select stack, dir_ip,campo, fecha, valor
from interfaces_hst
where (to_char(fecha,'yyyymm') between
to_char(add_months(sysdate,-2),'yyyymm') and
to_char(add_months(sysdate,-1),'yyyymm'))
and campo = 'ifOutOctets'
order by stack,dir_ip,campo,fecha asc)
order by stack,dir_ip,campo,fecha asc)
order by stack,dir_ip,campo,fecha asc)
where to_char(fecha,'yyyymm')= to_char(add_months(sysdate,-
1),'yyyymm')
group by stack, dir_ip,campo, trunc(fecha)
order by stack,dir_ip,trunc(fecha) asc;
```

```
commit;  
end interfa_sum_dia;
```

## mlt.sumariza.interfa\_sum\_hora

```
procedure interfa_sum_hora is  
begin  
insert into interfaces_sum_hora  
select stack, dir_ip,campo, trunc(fecha,'hh24'),sum(diferencia)  
from  
(select stack, dir_ip,campo, fecha, valor,ant,  
case when (to_number(valor) < to_number(ant))  
then 0  
else to_number(valor-ant)  
end diferencia  
from  
(select stack, dir_ip,campo, fecha,valor ,(nvl(lag(valor) over  
(partition by stack, dir_ip,campo order by fecha),0)) ant  
from  
(select stack, dir_ip,campo, fecha, valor  
from interfaces_hst  
where (to_char(fecha,'yyyymm') between  
to_char(add_months(sysdate,-2),'yyyymm') and  
to_char(add_months(sysdate,-1),'yyyymm'))  
and campo = 'ifInOctets'  
order by stack,dir_ip,campo,fecha asc)  
order by stack,dir_ip,campo,fecha asc)  
order by stack,dir_ip,campo,fecha asc)  
where to_char(fecha,'yyyymm')= to_char(add_months(sysdate,-  
1),'yyyymm')  
group by stack, dir_ip,campo, trunc(fecha,'hh24')  
order by stack,dir_ip,trunc(fecha,'hh24') asc;  
commit;  
insert into interfaces_sum_hora  
select stack, dir_ip,campo, trunc(fecha,'hh24'), sum(diferencia)  
from  
(select stack, dir_ip,campo, fecha, valor,ant,  
case when (to_number(valor) < to_number(ant))  
then 0  
else to_number(valor-ant)  
end diferencia  
from  
(select stack, dir_ip,campo, fecha,valor ,(nvl(lag(valor) over  
(partition by stack, dir_ip,campo order by fecha),0)) ant  
from  
(select stack, dir_ip,campo, fecha, valor  
from interfaces_hst  
where (to_char(fecha,'yyyymm') between  
to_char(add_months(sysdate,-2),'yyyymm') and  
to_char(add_months(sysdate,-1),'yyyymm'))  
and campo = 'ifOutOctets'  
order by stack,dir_ip,campo,fecha asc)  
order by stack,dir_ip,campo,fecha asc)  
order by stack,dir_ip,campo,fecha asc)  
where to_char(fecha,'yyyymm')= to_char(add_months(sysdate,-  
1),'yyyymm')  
group by stack, dir_ip,campo, trunc(fecha,'hh24')  
order by stack,dir_ip,trunc(fecha,'hh24') asc;  
commit;  
end interfa_sum_hora;  
end sumariza;
```

## CAPÍTULO V. PROGRAMACION

### V.1. Introducción.

Aunque como comentamos en el **capítulo III Diseño de la Arquitectura** el PHP será el lenguaje principal de desarrollo de la aplicación vamos a describir brevemente tanto este como las posibles alternativas. Realizaremos un repaso entre las distintas opciones que tenemos en la actualidad tomando como única condición que se trate de lenguajes que nos permitan generar contenidos dinámicos necesarios para el proyecto que queremos abordar.

#### *PHP*

Es una tecnología de código abierto que está actualmente en pleno crecimiento. PHP es el acrónimo de *Personal Home Page* y utiliza una sintaxis semejante al lenguaje C. Es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas. Es usado principalmente en interpretación del lado del servidor (server-side scripting) pero actualmente puede ser utilizado desde una interfaz de línea de comandos o en la creación de otros tipos de programas incluyendo aplicaciones con interfaz gráfica usando las bibliotecas Qt o GTK+.

#### Ventajas

- Es un lenguaje multiplataforma.
- Completamente orientado a la web.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando la enorme cantidad de módulos (llamados ext's o extensiones).
- Posee una amplia documentación en su página oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Se permite usar las técnicas de Programación Orientada a Objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.

- No se requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Manejo de excepciones (desde PHP5).
- Si bien en PHP no se obliga a quien lo usa a seguir una determinada metodología a la hora de programar (muchos otros lenguajes tampoco lo hacen), aun estando dirigido a alguna en particular, el programador puede aplicar en su trabajo cualquier técnica de programación y/o desarrollo que le permita escribir código ordenado, estructurado y manejable. Un ejemplo de esto son los desarrollos que en PHP se han hecho del patrón de diseño Modelo Vista Controlador (o MVC), que permiten separar el tratamiento y acceso a los datos, la lógica de control y la interfaz de usuario en tres componentes independientes.

#### Desventajas

- La ofuscación de código es la única forma de ocultar los fuentes.

### Common Gateway interface (CGI)

La primera forma de creación de contenido dinámico en páginas Web fue a través del mecanismo Common Gateway Interface (CGI), a través del cual los servidores Web pueden pasar información a páginas externas, que serán ejecutadas en el servidor Web para generar respuestas en tiempo de ejecución. El lenguaje Perl es el más utilizado para escribir este tipo de programas, aunque se puede utilizar cualquier lenguaje que genere programas que puedan ser invocados por el servidor Web, por ejemplo, cualquier lenguaje script soportado por el sistema operativo en donde esté corriendo el servidor Web, o un programa escrito en C y compilado, o una aplicación Java.

La tecnología CGI no está exenta de ineficiencias que la hacen desaconsejable en aplicaciones medianamente complejas. Su inconveniente más importante deriva de su propia filosofía, la ejecución de programas externos para la generación de la respuesta al cliente. Cada petición genera un nuevo proceso externo, lo cual, en servidores que atienden a muchas peticiones simultáneas, es una sobrecarga difícil de soportar.

## FASTCGI

Existe otra opción que es el FastCGI. Es una alternativa al CGI estándar, cuya diferencia radica principalmente en el hecho de que el servidor crea un único proceso persistente por cada programa FastCGI en lugar de por cada solicitud del cliente.

Aunque FastCGI es un paso en la dirección correcta, sigue teniendo problemas con la proliferación de procesos, ya que en todo momento existe al menos un proceso activo por cada programa FastCGI. Para manejar solicitudes concurrentes, habría que mantener un estanque de procesos, uno por cada solicitud. Considerando que cada uno de estos procesos puede estar ejecutando el intérprete de Perl, este modelo no parece tan distinto del CGI estándar. Una solución que FastCGI ofrece para este problema es su habilidad para distribuir todos estos procesos entre múltiples servidores.

## ColdFusion

La tecnología ColdFusion, creada por Allaire, se basa en una serie de etiquetas HTML que soportan gran variedad de acciones para la generación de contenido dinámico. Estas etiquetas permiten, por ejemplo, realizar consultas a bases de datos, y mantienen una consistencia única con las etiquetas HTML del resto de la página.

## JavaScript

Es un lenguaje de script, interpretado, orientado a objeto, que permite introducir interactividad en documentos HTML y tiene la ventaja de que no necesita ser transmitido hacia el servidor, verificado y devuelto. Éste es ejecutado en el navegador del usuario. Los scripts de Javascript pueden ser introducidos dentro de sus páginas de HTML. Con Javascript se puede dar respuesta a eventos iniciados por el usuario, eventos tales como la entrada de una forma o algún enlace. Las entradas son verificadas por la aplicación cliente y pueden ser transmitidas después de esto. Permite efectuar cálculos, efectos especiales, verificar formas, crear juegos, personalizar la gráfica, crear password de seguridad, y mucho más. Realiza el manejo de muchos recursos Web.



## Server-Side JavaScript

Esta tecnología permite utilizar javaScript ejecutándose en el servidor web. La tecnología Server-Side JavaScript (SSJS) incorpora otras características al lenguaje, como es el soporte para correo electrónico, control de sesiones e interoperabilidad con Java ejecutándose en el servidor a través de la tecnología LiveWire de Netscape.

El inconveniente de esta tecnología reside en su incompatibilidad, es decir, que aunque puede ejecutarse en cualquier plataforma o sistema operativo, al igual que Java, es una tecnología desarrollada específicamente para los servidores Netscape, o iPlanet.

## Active Server Pages (ASP)

El ASP está basado en la inclusión de etiquetas al estilo ColdFusion, pero permite utilizar un lenguaje script, por defecto VBScript, un subconjunto de Visual Basic de Microsoft. Puede combinar código HTML, scripts y componentes ActiveX del servidor para crear soluciones dinámicas y muy potentes para la Web. Microsoft introdujo ésta tecnología como parte del Internet Information Server (IIS). El script por defecto es el VBScript, pero existe otra diversidad de lenguajes que pueden ser utilizados como lo es Perl, JScript, etc.

El ASP es una tecnología dinámica funcionando del lado del servidor, lo que significa que cuando el usuario solicita un documento ASP, las instrucciones de programación dentro del script son ejecutadas para enviar al navegador únicamente el código HTML resultante.

La ventaja principal de las tecnologías dependientes del servidor radica en la seguridad que tiene el programador sobre su código, ya que éste se encuentra únicamente en los archivos del servidor que al ser solicitado a través del web, es ejecutado, por lo que los usuario no tienen acceso más que a la página resultante en su navegador.

Sus funciones principales están el acceso a bases de datos, envío de correo electrónico, creación dinámica de gráficos y otros. Muchas cosas que se pueden realizar por medio de CGI pueden ser realizadas con esta tecnología. Esto es debido a que el ASP es tan eficiente como escribir código directamente en la interfaz de aplicación del servidor, con la ventaja de que es más eficiente que el CGI (que depende de un compilador) ya que el ASP corre como un servicio en el servidor,

tomando ventaja de la arquitectura de multitareas. El único inconveniente con las ASP es que solamente trabajan sobre plataformas Windows 9x, NT, XP, Windows 7 y Linux.

## Servlets

La tecnología Servlet proporciona las mismas ventajas del lenguaje Java en cuanto a portabilidad (“write once, run anywhere”) y seguridad, ya que un servlet es una clase de Java igual que cualquier otra, y por tanto tiene en ese sentido todas las características del lenguaje. Esto es algo de lo que carecen los programas CGI, ya que hay que compilarlos para el sistema operativo del servidor y no disponen en muchos casos de técnicas de comprobación dinámica de errores en tiempo de ejecución.

Otra de las principales ventajas de los servlets con respecto a los programas CGI, es la del rendimiento y esto a pesar de que Java no es un lenguaje particularmente rápido. Mientras que es necesario cargar los programas CGI tantas veces como peticiones de servicio existan por parte de los clientes, los servlets, una vez que son llamados por primera vez, quedan activos en la memoria del servidor hasta que el programa que controla el servidor los desactiva. De esta manera se minimiza en gran medida el tiempo de respuesta.

Además, los servlets se benefician de la gran capacidad de Java para ejecutar métodos en ordenadores remotos, para conectar con bases de datos, para la seguridad en la información, etc. Se podría decir que las clases estándar de java ofrecen resueltos muchos problemas que con otros lenguajes tiene que resolver el programador.

Además de las características indicadas, los servlets tienen estas otras:

- Son independientes del servidor utilizado y de su sistema operativo, lo que quiere decir que a pesar de estar escritos en Java, el servidor puede estar escrito en cualquier lenguaje de programación, obteniéndose exactamente el mismo resultado que si lo estuviera en Java.
- Los servlets pueden llamar a otros servlets, e incluso a métodos concretos de otros servlets. De esta forma se puede distribuir de forma más eficiente el trabajo a realizar. Se podría tener un servlet encargado

de la interacción con los clientes y que llamara a otro servlet para que a su vez se encargara de la comunicación con una base de datos. De igual forma, los servlets permiten redireccionar peticiones de servicios a otros servlets en la misma máquina o en una remota.

- Los servlets pueden obtener fácilmente información acerca del cliente (la permitida por el protocolo http), tal como su dirección IP, el puerto que se utiliza en la llamada, el método utilizado (GET, POST), etc.
- Permiten además la utilización de cookies y sesiones, de forma que se puede guardar información específica acerca de un usuario determinado, personalizado de esta forma la interacción cliente-servidor. Una clara aplicación es mantener la sesión con un cliente.
- Los servlets pueden actuar como enlace entre el cliente y una o varias bases de datos en arquitecturas cliente-servidor de tres capas.
- Pueden realizar tareas de proxy para un applet. Debido a las restricciones de seguridad, un applet no puede acceder directamente por ejemplo a un servidor de datos localizado en cualquier máquina remota, pero el servlet sí puede hacerlo desde su parte.
- Al igual que los programas CGI, los servlets permiten la generación dinámica de código HTML dentro de una propia página HTML. Así, pueden emplearse servlets para la creación de contadores, banners, etc.

## JavaServer Pages

De lo escrito hasta ahora se desprende que la incorporación de contenido en dinámico en un sitio Web siempre lleva consigo algún tipo de programación para indicar cómo debe generarse ese contenido dinámico. Sin embargo, la programación requiere unos conocimientos que no todo el mundo posee, tiende a resultar cara y difícil de mantener, por lo que una de las metas en la creación de contenido dinámico es minimizar la necesidad de programación y, en último caso, separar la programación de la presentación del contenido. Combinando estos objetivos, el uso de Java y la utilización de etiquetas, la tecnología JavaServer Pages (JSP) es el resultado creado por Sun Microsystems.

Aunque los servlets pueden ser un programa completo para la generación de respuesta atendiendo a peticiones del cliente, la tarea de la generación de contenido dinámico debe ser separada en dos partes, para facilitar la programación y reducir en

lo posible el coste de creación y mantenimiento. Las dos partes que intervienen en la generación de contenidos dinámicos son entonces:

- Lógica de negocio, creación de contenidos, que controla la relación entre la entrada, los algoritmos y la salida.
- Lógica de presentación, presentación de contenidos o diseño gráfico, que determina la forma en que se va a presentar la información al usuario.

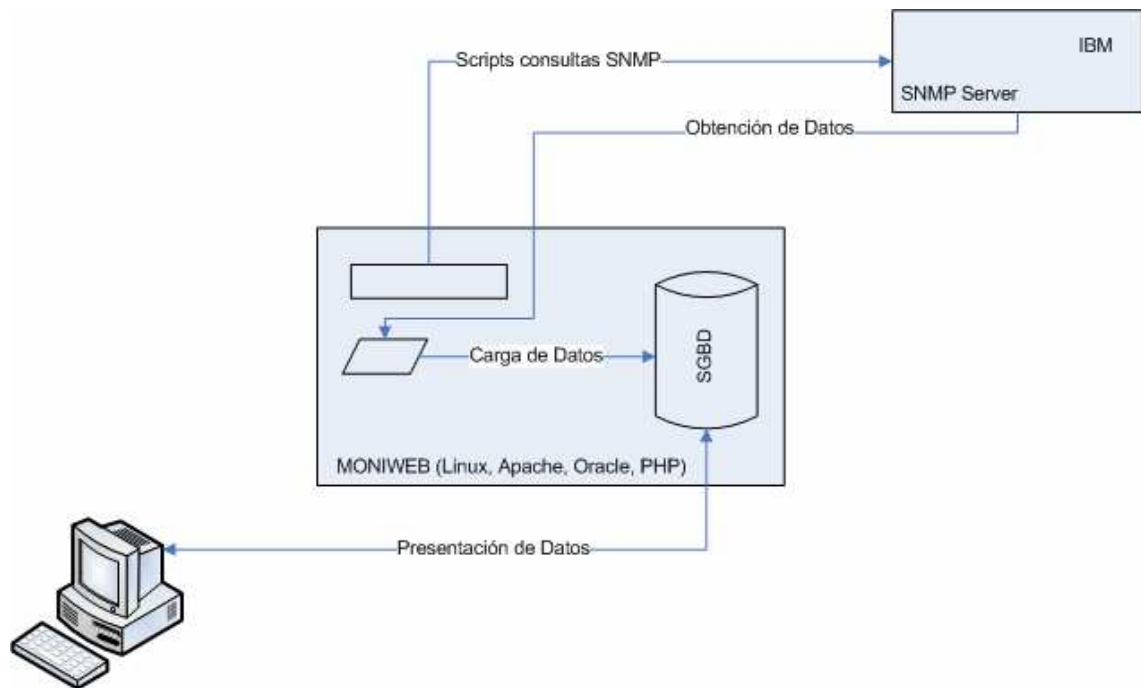
En este escenario, la lógica de negocio puede ser controlada desde JavaBean y la lógica de presentación puede ser manejada a través de la tecnología JSP, mientras que los servlets se encargan del control del protocolo http.

La tecnología JSP es un híbrido, porque por un lado soporta el código embebido en sus páginas, al igual que ASP, PHP o SSJS; pero por otro lado, también permite el uso de etiquetas que interactúan con objetos Java en el servidor, al igual que ColdFusion.

Con este modelo híbrido, la tecnología JSP proporciona muchas ventajas. Los desarrolladores pueden ofrecer etiquetas personalizadas que los diseñadores de páginas pueden utilizar mediante sintaxis semejante a las etiquetas HTML que ya conocen.

Como el motor JSP es capaz de compilar la página JSP bajo demanda, el autor de la página puede realizar actualizaciones fácilmente. Las páginas JSP pueden proporcionar acceso a componentes JavaBeans que encapsulan la lógica de negocio, o programación, acceso a datos, etc. Estos componentes, una vez escritos, son portables entre plataformas y servidores. La reutilización de los componentes ya existentes acelera el desarrollo de nuevas aplicaciones.

Los diseñadores de páginas web pueden modificar y editar la parte estática de la página tantas veces como deseen, si afectar a la lógica de la aplicación. Del mismo modo que los desarrolladores pueden introducir cambios en los algoritmos a nivel de un componente JavaBean sin tener que editar cada una de las páginas que utilice ese componente.



## V.2 Obtención de Datos

El proceso de obtención de los datos ha sido uno de los que me ha llevado más tiempo, no por la complejidad en la obtención de los datos, sino por el estudio y decisión de que datos eran significativos para su posterior representación y de la elección del mejor sistema para la extracción de los mismos. De hecho algunos datos que no se han podido obtener directamente han necesitado del desarrollo de cierto código.

Basándome en mi experiencia como Técnico de Sistemas en entornos Mainframe de IBM y con el fin de utilizar el mínimo de recursos (dado el alto coste económico de los mismos) la opción de utilizar las consultas SNMP se presentó como la mejor frente a otras, como la ejecución de comandos en HOST, su almacenamiento en ficheros y su posterior extracción.

El Host IBM dispone de un servidor SNMP que cargado con las MIB's apropiadas nos proporciona toda la información necesaria. A continuación se describen los scripts que extraen dicha información y la almacenan directamente en ficheros para su posterior carga.

### moniweb\_MLTE.sh

Este script realizado en bash es el encargado de realizar la consulta de los datos de la partición EXPL y su almacenamiento. Los scripts de VTL3 (moniweb\_MLTV.sh) y CEIN (moniweb\_MLTN.sh) son idénticos por lo que no voy a proceder a describirlos. Está planificado en el crontab del usuario mlt para que se ejecute cada 4 minutos, ya que las variables MIB en el host IBM tienen ese mismo intervalo de refresco. Se lanza uno por cada partición en minutos dispares con el fin de no coincidir en el tiempo y optimizar los recursos de Moniweb.

```
#!/bin/ksh
# Elaboracion del primer registro con la fecha y la hora para el MLTE.
#
printf "hora " > /www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
date '+%Y%m%d-%H.%M' >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
# Informacion TCP de todo el STACK
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.6.5 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.6.6 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
```

```
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.6.7 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.6.8 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.6.9 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.6.10 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.6.11 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.6.12 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.6.14 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.6.15 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
#
# Informacion de los LISTENERS
#
snmpwalk -c password -O bq -m all IP1EPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.4 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.12 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.8 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.9 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.10 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.11 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.13 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
#
# Informacion de las conexiones
#
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.6.13.1.1 |
cut -b 31-256 >> /www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
#
# Informacion de los Interfaces
#
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.4.20.1.2 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.2.2.1.8 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.2.2.1.10 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.2.2.1.13 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.2.2.1.14 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.2.2.1.16 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
```

```
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.2.2.1.19 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1EPROC .1.3.6.1.2.1.2.2.1.20 >>
/www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
#
#Copiamos los datos al fichero bueno para que los cargue ORACLE
#
cp /www/htdocs/monitores/Datos/mlt/mlte_datos_ora.tmp
/www/htdocs/monitores/Datos/mlt/mlte_datos_oracle.txt
```

## moniweb\_MLTC.sh

Este script realizado en bash es el encargado de realizar la consulta de los datos de la partición SIAC y su almacenamiento. El script de SIAB (moniweb\_MLTB.sh) es idéntico por lo que no voy a proceder a describirlo. Está planificado en el crontab del usuario mlt para que se ejecute cada 4 minutos, ya que las variables MIB en el host IBM tienen ese mismo intervalo de refresco. Se lanza uno por cada partición en minutos dispares con el fin de no coincidir en el tiempo y optimizar los recursos de Moniweb.

```
#!/bin/ksh
#
# Elaboración del primer registro con la fecha y la hora para el MLTC.
#
printf "hora " > /www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
date '+%y%m%d-%H.%M' >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
# Informacion TCP de todo el STACK
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.6.5 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.6.6 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.6.7 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.6.8 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.6.9 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.6.10 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.6.11 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.6.12 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.6.14 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.6.15 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
#
# Informacion de los LISTENERS
#
snmpwalk -c password -O bq -m all IP1CPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.4 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
```



```
snmpwalk -c password -O bq -m all IP1CPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.12 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.8 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.9 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.10 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.11 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC
.1.3.6.1.4.1.2.6.19.2.2.7.31.1.13 | cut -b 151-256 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
#
# Informacion de las conexiones
#
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.6.13.1.1 |
cut -b 31-256 >> /www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
#
# Informacion de Sermepa
#
snmpwalk -c password -O bq -m all IP1CPROC
.1.3.6.1.4.1.2.6.19.2.2.7.1.1.2 | cut -b 143-250 | grep 10.99.5.32 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1BPROC .1.3.6.1.2.1.6.13.1.1 | cut
-b 31-125 | grep 10.99.5.32 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1BPROC
.1.3.6.1.4.1.2.6.19.2.2.7.1.1.2 | cut -b 143-250 | grep 10.99.5.32 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
#
# Informacion de los Interfaces
#
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.4.20.1.2 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.2.2.1.8 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.2.2.1.10 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.2.2.1.13 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.2.2.1.14 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.2.2.1.16 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.2.2.1.19 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
snmpwalk -c password -O bq -m all IP1CPROC .1.3.6.1.2.1.2.2.1.20 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
#
# Informacion de las VIPA dinámicas
#
snmpwalk -c password -O bq -m all IP1CPROC
.1.3.6.1.4.1.2.6.19.2.2.11.1.1.5 | cut -b 131-200 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
```

```
snmpwalk -c password -O bq -m all IP1CPROC
.1.3.6.1.4.1.2.6.19.2.2.11.5.1.5 | cut -b 147-222 >>
/www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
#
#Copiamos los datos al fichero bueno para que los cargue ORACLE
#
cp /www/htdocs/monitores/Datos/mlt/mltc_datos_ora.tmp
/www/htdocs/monitores/Datos/mlt/mltc_datos_oracle.txt
```

## V.3 Carga de Datos

### mltb\_snmp\_oracle.php

Una vez obtenidos los datos y almacenados en ficheros procedemos a la carga de los mismos en nuestras tablas. Para ello se han desarrollado varios programas muy similares en PHP que discriminan los datos, realizan los cálculos necesarios, montan las query's y las lanzan contra la base de datos ORACLE. Este programa como el resto de scripts está planificado en el crontab del usuario mlt para que se ejecute cada 4 minutos. Se lanza uno por cada fichero en minutos dispares con el fin de no coincidir en el tiempo y optimizar los recursos de Moniweb. Los programas desarrollados son **mlt\_snmp\_oracle.php**, **mltv\_snmp\_oracle.php**, **mltc\_snmp\_oracle.php**, **mltb\_snmp\_oracle.php** y **mltn\_snmp\_oracle.php**.

A continuación se incluye como "única" muestra de programación PHP el programa mlt\_snmp\_oracle.php.

```
#!/usr/local/bin/php
<?php
    include('/www/htdocs/monitores/lib/entorno.php');
    $Result=0;
    if ($_SESSION['conn'] =
@oci_connect("usuario","password","database"))
    {
        $Result=0;
    }
    else
    {
        $e = oci_error();
        $Msg=$e['message'];

        $_SESSION['mensaje'] = $Msg;
        $_SESSION['resultado_oracle'] = substr($Msg,4,5);
        $Result = -1;
    } // fin del logon

    /* require('/www/htdocs/monitores/lib/func_ora.php'); */
    /* lectura del fichero de con los datos del TCPIP del IBM */

$nombreArchivo="/www/htdocs/monitores/Datos/mlt/mlte_datos_oracle.txt"
;
$f=fopen($nombreArchivo,"r");
```

```
$tamTotal=filesize($nombreArchivo);
$linea=fread($f,$tamTotal);
fclose($f);
/* bucle para leer las líneas e interpretarlas */
for ($contador=1;strlen($linea)<>0;$contador++) {
    $cont=strpos($linea,"\n");
    /* he cambiado $cont-1 por $cont */
    $temp = substr($linea, 0,$cont);
    /* obtenemos el primer dato para usarlo como referencia en el
array $datos */
    $scanned = explode (" ", $temp);
    list($lit1,$dat)=$scanned;
    $lit1=trim($lit1);
    $dat=trim($dat);
    $matriz_var[$lit1]=$dat;
    /* eliminamos lo tratado */
    if (strlen($linea)<=$cont) {
        $linea="";
    } else {
        $linea=substr($linea,$cont+1,$tamTotal);
    }
}

/* leer de la tabla de alarmas_activas para obtener el estado
anterior */
require_once('mlt_select_oracle.php');
$sentencia="select stack, puerto, estado from
mlt.alarmas_activas where stack='EXPL'";
select_de_oracle($sentencia,&$matriz_select);
for ($i=0;$i < count($matriz_select);$i=$i+3){

    $_SESSION['alarma_puerto_activa'][trim($matriz_select[$i+1])][trim($matriz_select[$i+2])]=trim($matriz_select[$i+3]);
}
/* Fin de lectura del fichero de datos */
/* voy a insertar en oracle */
$stack='EXPL';

/* Voy a procesar primero los tcpConnState para hacer un único
insert y olvidarme de los update */
$numero_total_sockets=0;

foreach($matriz_var as $campo => $dato) {
    $marca0 = 'hora';
    $esmarca0 = strpos($campo,$marca0);
    $marca6 = 'tcpConnState';
    $esmarca6 = strpos($campo,$marca6);
    if ($esmarca0 != false):
        $fecha=str_replace("-", " ", $dato);
    elseif ($esmarca6 != false):
        $numero_total_sockets=$numero_total_sockets+1;
    else:
        nop;
    endif;
}

$tabla='mlt.stack_tcp';
$campo_1 = 'xsockets';
$valor_1=$numero_total_sockets;
```

```

$valores = " ('".$stack."','".$campo_1."',
to_date('".$fecha."','".$ymmd hh24.mi')."','".$valor_1."");
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
unset($mensajes);
foreach($matriz_var as $campo => $dato) {
    $marca0 = 'hora';
    $esmarca0 = strpos($campo,$marca0);
    $marca1 = 'tcp.tcp';
    $esmarca1 = strpos($campo,$marca1);
    $marca2 = 'ibmMvsTcpListenerLocalPort';
    $esmarca2 = strpos($campo,$marca2);
    $marca4 = 'ipAddrEntry';
    $esmarca4 = strpos($campo,$marca4);
    if ($esmarca0 != false):
        $fecha=$dato;
    elseif ($esmarca1 != false):
        procesotcp($stack,$fecha,$campo,$dato);
    elseif ($esmarca2 != false):

procesoibmMvsTcpListenerLocalPort($stack,$fecha,$campo,$dato,$matriz_v
ar,$matriz_established,$matriz_finWait2,$matriz_timeWait,$matriz_close
Wait,&$mensajes);
    elseif ($esmarca4 != false):
        procesoipAddr($stack,$fecha,$campo,$dato,$matriz_var);
    else:
        nop;
    endif;
}
$pos=0;
if (isset($mensajes)) {
    foreach ($mensajes as $k => $d) {
        //enviar mensaje para cada error
        $t="insert into moni.avisos (monitor, tipo, fecha,
cabecera, texto) ";
        $t.="values ('MLT','1',to_date('".date('d/m/Y
H:i:s')."','".$dd/mm/yyyy hh24:mi:ss'),'Puerto ".$k."', ";
        $t.="','".$mensajes[$k]."'");
        $sentencia=$t;
        ejecutar_insert($sentencia,0);
    }
}

foreach ($_SESSION['alarma_puerto_activa'] as $k => $d) {
    $sentencia="delete from mlt.alarmas_activas where stack='".$k."'";
    ejecutar_insert($sentencia,0);
    foreach ($_SESSION['alarma_puerto_activa'][$k] as $k1 => $d1) {
        $sentencia="insert into mlt.alarmas_activas (stack, puerto,
estado) values
('".$k."','".$k1."','".$_SESSION['alarma_puerto_activa'][$k][$k1]."'")"
;
        ejecutar_insert($sentencia,0);
    }
}
oci_commit($_SESSION['conn']);
oci_close($_SESSION['conn']);
?>
<?php
function procesotcp($stack,$fecha,$campo='', $dato='') {
    $tabla='mlt.stack_tcp';
    $scanned = explode (".", $campo);

```

```
list($lit1,$lit2,$dat)=$scanned;
$campo_1=$lit2;
$valores=$dato;
$valores = " ('".$stack."','".$campo_1."',
to_date('".$fecha."','".$ymmd hh24.mi')','".$valor_1.")";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
}
function
procesoibmMvsTcpListenerLocalPort($stack,$fecha,$campo='', $dato='', $ma
triz_var,$matriz_established,$matriz_finWait2,$matriz_timeWait,$matriz
_closeWait) {
$tabla='mlt.sockets';
$scanned= explode (".", $campo);
list($lit1,$dat)=$scanned;
$puntero=$dat;
$campo='ibmMvsTcpListenerResourceName.'.$puntero;
$nombre = $matriz_var[$campo];
$puerto = $dato;
/* Número de Accepts Counts */
$campo_1 = 'AcceptCount';
$val_dato='ibmMvsTcpListenerAcceptCount.'.$puntero;
$valor_1=$matriz_var[$val_dato];
$valores = " ('".$stack."','".$";
$valores = $valores.$nombre."','".$";
$valores = $valores.$puerto."','".$";
$valores = $valores.$campo_1."','".$";
$valores = $valores." to_date('".$fecha."','".$ymmd hh24.mi')",";
$valores = $valores.$valor_1.")";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* Número de ExceedBacklog */
$campo_1 = 'ExceedBacklog';
$val_dato='ibmMvsTcpListenerExceedBacklog.'.$puntero;
$valor_1=$matriz_var[$val_dato];
$valores = " ('".$stack."','".$";
$valores = $valores.$nombre."','".$";
$valores = $valores.$puerto."','".$";
$valores = $valores.$campo_1."','".$";
$valores = $valores." to_date('".$fecha."','".$ymmd hh24.mi')",";
$valores = $valores.$valor_1.")";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* Número de CurrBacklog */
$campo_1 = 'CurrBacklog';
$val_dato='ibmMvsTcpListenerCurrBacklog.'.$puntero;
$valor_1=$matriz_var[$val_dato];
$valores = " ('".$stack."','".$";
$valores = $valores.$nombre."','".$";
$valores = $valores.$puerto."','".$";
$valores = $valores.$campo_1."','".$";
$valores = $valores." to_date('".$fecha."','".$ymmd hh24.mi')",";
$valores = $valores.$valor_1.")";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
if ($valor_1>20) {
    if
(isset($_SESSION['alarma_puerto_activa'])(trim($stack))(trim($puerto)))
===false) {
```

```
$_SESSION['alarma_puerto_activa'][trim($stack)][trim($puerto)]=  
';  
}  
if  
($_SESSION['alarma_puerto_activa'][trim($stack)][trim($puerto)]!='ENV'  
) {  
    //mensajito  
    $mensajes[$puerto]=$stack.": Cola de Backlog del puerto  
    ".$puerto." [".$nombre."] con valor ".$valor_1;  
    echo $mensajes[$puerto].'<br>';  
  
    $_SESSION['alarma_puerto_activa'][trim($stack)][trim($puerto)]=  
    ENV';  
}  
} else {  
    $_SESSION['alarma_puerto_activa'][trim($stack)][trim($puerto)]=  
    ';  
}  
/* Número de MaxBacklog */  
$campo_1 = 'MaxBacklog';  
$val_dato='ibmMvsTcpListenerMaxBacklog.'.$puntero;  
$valor_1=$matriz_var[$val_dato];  
$valores = " ( ".$stack."',";  
$valores = $valores.$nombre."',";  
$valores = $valores.$puerto."',";  
$valores = $valores.$campo_1."',";  
$valores = $valores." to_date('".$fecha."', 'yymmdd hh24.mi') ,";  
$valores = $valores.$valor_1.")";  
$sentencia = "insert into ".$tabla." values".$valores;  
ejecutar_insert($sentencia,0);  
/* Número de Established */  
$campo_1 = 'Established';  
/* Modificación para sacar datos de Listeners en SharePoint */  
$val_dato='ibmMvsTcpListenerCurrConns.'.$puntero;  
$valor_1=$matriz_var[$val_dato];  
/* $valor_1='0';  
if (isset($matriz_established[$puerto])) {  
    $valor_1=$matriz_established[$puerto];  
} */  
$valores = " ( ".$stack."',";  
$valores = $valores.$nombre."',";  
$valores = $valores.$puerto."',";  
$valores = $valores.$campo_1."',";  
$valores = $valores." to_date('".$fecha."', 'yymmdd hh24.mi') ,";  
$valores = $valores.$valor_1.")";  
$sentencia = "insert into ".$tabla." values".$valores;  
ejecutar_insert($sentencia,0);  
/* Número de FinWait2 */  
$campo_1 = 'FinWait2';  
$valor_1='0';  
if (isset($matriz_finWait2[$puerto])) {  
    $valor_1=$matriz_finWait2[$puerto];  
}  
$valores = " ( ".$stack."',";  
$valores = $valores.$nombre."',";  
$valores = $valores.$puerto."',";  
$valores = $valores.$campo_1."',";  
$valores = $valores." to_date('".$fecha."', 'yymmdd hh24.mi') ,";  
$valores = $valores.$valor_1.")";  
$sentencia = "insert into ".$tabla." values".$valores;
```

```
ejecutar_insert($sentencia,0);
/* Número de TimeWait */
$campo_1 = 'TimeWait';
$valor_1='0';
if (isset($matriz_timeWait[$puerto])) {
    $valor_1=$matriz_timeWait[$puerto];
}
$valores = " ('".$stack."','".$;
$valores = $valores.$nombre."','".$;
$valores = $valores.$puerto."','".$;
$valores = $valores.$campo_1."','".$;
$valores = $valores." to_date('".$fecha."','".$ymmd hh24.mi'),'";
$valores = $valores.$valor_1."";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);

/* Número de closeWait */
$campo_1 = 'CloseWait';
$valor_1='0';
if (isset($matriz_closeWait[$puerto])) {
    $valor_1=$matriz_closeWait[$puerto];
}
$valores = " ('".$stack."','".$;
$valores = $valores.$nombre."','".$;
$valores = $valores.$puerto."','".$;
$valores = $valores.$campo_1."','".$;
$valores = $valores." to_date('".$fecha."','".$ymmd hh24.mi'),'";
$valores = $valores.$valor_1."";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
}
function
procesoconexionmentirosa($stack,$fecha,$campo='', $point,$name,$port1)
{
    $tabla='mlt.sockets';
    $puntero=$point;
    $campo='ibmMvsTcpListenerResourceName.'.$puntero;
    $nombre = $name;
    $puerto = $port1;
    /* Mentira Número de Accepts Counts */
    $campo_1 = 'AcceptCount';
    $val_dato='ibmMvsTcpListenerAcceptCount.'.$puntero;
    $valor_1='0';
    $valores = " ('".$stack."','".$;
    $valores = $valores.$nombre."','".$;
    $valores = $valores.$puerto."','".$;
    $valores = $valores.$campo_1."','".$;
    $valores = $valores." to_date('".$fecha."','".$ymmd hh24.mi'),'";
    $valores = $valores.$valor_1."";
    $sentencia = "insert into ".$tabla." values".$valores;
    ejecutar_insert($sentencia,0);
    /* Número de ExceedBacklog */
    $campo_1 = 'ExceedBacklog';
    $val_dato='ibmMvsTcpListenerExceedBacklog.'.$puntero;
    $valor_1='0';
    $valores = " ('".$stack."','".$;
    $valores = $valores.$nombre."','".$;
    $valores = $valores.$puerto."','".$;
    $valores = $valores.$campo_1."','".$;
    $valores = $valores." to_date('".$fecha."','".$ymmd hh24.mi'),'";
    $valores = $valores.$valor_1."";
```

```
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* Número de CurrBacklog */
$campo_1 = 'CurrBacklog';
$val_dato='ibmMvsTcpListenerCurrBacklog.'. $puntero;
$valor_1='0';
$valores = " ('".$stack."' , '";
$valores = $valores.$nombre."' , '";
$valores = $valores.$puerto."' , '";
$valores = $valores.$campo_1."' , '";
$valores = $valores." to_date('".$fecha."' , 'yymmdd hh24.mi') , ";
$valores = $valores.$valor_1.")";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* Número de MaxBacklog */
$campo_1 = 'MaxBacklog';
$val_dato='ibmMvsTcpListenerMaxBacklog.'. $puntero;
$valor_1='0';
$valores = " ('".$stack."' , '";
$valores = $valores.$nombre."' , '";
$valores = $valores.$puerto."' , '";
$valores = $valores.$campo_1."' , '";
$valores = $valores." to_date('".$fecha."' , 'yymmdd hh24.mi') , ";
$valores = $valores.$valor_1.")";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* Número de Established */
$campo_1 = 'Established';
$valor_1='1';
$valores = " ('".$stack."' , '";
$valores = $valores.$nombre."' , '";
$valores = $valores.$puerto."' , '";
$valores = $valores.$campo_1."' , '";
$valores = $valores." to_date('".$fecha."' , 'yymmdd hh24.mi') , ";
$valores = $valores.$valor_1.")";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* Número de FinWait2 */
$campo_1 = 'FinWait2';
$valor_1='0';
$valores = " ('".$stack."' , '";
$valores = $valores.$nombre."' , '";
$valores = $valores.$puerto."' , '";
$valores = $valores.$campo_1."' , '";
$valores = $valores." to_date('".$fecha."' , 'yymmdd hh24.mi') , ";
$valores = $valores.$valor_1.")";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* Número de TimeWait */
$campo_1 = 'TimeWait';
$valor_1='0';
$valores = " ('".$stack."' , '";
$valores = $valores.$nombre."' , '";
$valores = $valores.$puerto."' , '";
$valores = $valores.$campo_1."' , '";
$valores = $valores." to_date('".$fecha."' , 'yymmdd hh24.mi') , ";
$valores = $valores.$valor_1.")";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);

/* Número de closeWait */
```



```
$campo_1 = 'CloseWait';
$valor_1='0';
$valores = " ('".$stack."','".$;
$valores = $valores.$nombre."','".$;
$valores = $valores.$puerto."','".$;
$valores = $valores.$campo_1."','".$;
$valores = $valores." to_date('".$fecha."','".$ymmd hh24.mi'),'";
$valores = $valores.$valor_1."");
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);

}
function procesoipAddr($stack,$fecha,$campo='', $dato='', $matriz_var) {
$ip='0.0.0.0';
$ip0 = '10.100.0.1';
$esip0 = strpos($campo,$ip0);
if ($esip0 !== false) {
    $ip='VI1.'.$ip0;
}
$ip1 = '10.254.32.45';
$esip1 = strpos($campo,$ip1);
if ($esip1 !== false) {
    $ip='IF1.'.$ip1;
}
$ip2 = '10.254.32.48';
$esip2 = strpos($campo,$ip2);
if ($esip2 !== false) {
    $ip='IF2.'.$ip2;
}
$ip3 = '10.249.32.46';
$esip3 = strpos($campo,$ip3);
if ($esip3 !== false) {
    $ip='IF1.'.$ip3;
}
$ip4 = '10.249.32.47';
$esip4 = strpos($campo,$ip4);
if ($esip4 !== false) {
    $ip='IF2.'.$ip4;
}
if ($ip !== '0.0.0.0') {
    $tabla='mlt.interfaces';
    $direccion=$ip;
    $puntero=$dato;
    /* ifOperStatus */
    $campo_1='ifOperStatus';
    $campo='interfaces.ifTable.ifEntry.ifOperStatus.'.$dato;
    $valor_1=$matriz_var[$campo];
    $valores = " ('".$stack."','".$;
    $valores = $valores.$direccion."','".$;
    $valores = $valores.$campo_1."','".$;
    $valores = $valores." to_date('".$fecha."','".$ymmd hh24.mi'),'";
    $valores = $valores.$valor_1."");
    $sentencia = "insert into ".$tabla." values".$valores;
    ejecutar_insert($sentencia,0);
    /* ifInOctets */
    $campo_1='ifInOctets';
    $campo='interfaces.ifTable.ifEntry.ifInOctets.'.$dato;
    $valor_1=$matriz_var[$campo];
    $valores = " ('".$stack."','".$;
    $valores = $valores.$direccion."','".$;
    $valores = $valores.$campo_1."','".$;
```

```
$valores = $valores." to_date('".$fecha."' , 'yymmdd hh24.mi') , '";
$valores = $valores.$valor_1." '";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* ifInDiscards */
$campo_1='ifInDiscards';
$campo='interfaces.ifTable.ifEntry.ifInDiscards.'.$dato;
$valor_1=$matriz_var[$campo];
$valores = " ( ' ".$stack." ' , '";
$valores = $valores.$direccion." ' , '";
$valores = $valores.$campo_1." ' , '";
$valores = $valores." to_date('".$fecha."' , 'yymmdd hh24.mi') , '";
$valores = $valores.$valor_1." '";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* ifInErrors */
$campo_1='ifInErrors';
$campo='interfaces.ifTable.ifEntry.ifInErrors.'.$dato;
$valor_1=$matriz_var[$campo];
$valores = " ( ' ".$stack." ' , '";
$valores = $valores.$direccion." ' , '";
$valores = $valores.$campo_1." ' , '";
$valores = $valores." to_date('".$fecha."' , 'yymmdd hh24.mi') , '";
$valores = $valores.$valor_1." '";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* ifOutOctets */
$campo_1='ifOutOctets';
$campo='interfaces.ifTable.ifEntry.ifOutOctets.'.$dato;
$valor_1=$matriz_var[$campo];
$valores = " ( ' ".$stack." ' , '";
$valores = $valores.$direccion." ' , '";
$valores = $valores.$campo_1." ' , '";
$valores = $valores." to_date('".$fecha."' , 'yymmdd hh24.mi') , '";
$valores = $valores.$valor_1." '";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* ifOutDiscards */
$campo_1='ifOutDiscards';
$campo='interfaces.ifTable.ifEntry.ifOutDiscards.'.$dato;
$valor_1=$matriz_var[$campo];
$valores = " ( ' ".$stack." ' , '";
$valores = $valores.$direccion." ' , '";
$valores = $valores.$campo_1." ' , '";
$valores = $valores." to_date('".$fecha."' , 'yymmdd hh24.mi') , '";
$valores = $valores.$valor_1." '";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
/* ifOutErrors */
$campo_1='ifOutErrors';
$campo='interfaces.ifTable.ifEntry.ifOutErrors.'.$dato;
$valor_1=$matriz_var[$campo];
$valores = " ( ' ".$stack." ' , '";
$valores = $valores.$direccion." ' , '";
$valores = $valores.$campo_1." ' , '";
$valores = $valores." to_date('".$fecha."' , 'yymmdd hh24.mi') , '";
$valores = $valores.$valor_1." '";
$sentencia = "insert into ".$tabla." values".$valores;
ejecutar_insert($sentencia,0);
}
}
```

```
function
procesoconexion($stack,$fecha,$campo='', $dato='', &$matriz_established,
&$matriz_finWait2, &$matriz_timeWait, &$matriz_closeWait) {
    $tabla='mlt.conexiones';
    $scanned = explode(".", $campo);
    list($lit1,$d1,$d2,$d3,$d4,$p5,$d6,$d7,$d8,$d9,$p10)=$scanned;
    $campo_1=$lit1;
    $valor_1="". $dato. "";
    $nombre=$d1.".".$d2.".".$d3.".".$d4;
    $puerto=$p5;
    $nombrepuerto=$d6.".".$d7.".".$d8.".".$d9.".".$p10;
    $nombre2=$d6.".".$d7.".".$d8.".".$d9;
    $puerto2=$p10;

    switch($nombrepuerto)
    {
        case("10.99.8.60.4527"):
            if ($dato == "established") {

procesoconexionmentirosa($stack,$fecha,'ibmMvsTcpListenerLocalPort.999
999 : INTEGER: 4527','999999','10.99.8.60','4527');
            }
            break;
        case("10.99.8.61.4527"):
            if ($dato == "established") {

procesoconexionmentirosa($stack,$fecha,'ibmMvsTcpListenerLocalPort.999
998 : INTEGER: 4527','999998','10.99.8.61','4527');
            }
            break;
        case("10.99.8.4.4209"):
            if ($dato == "established") {

procesoconexionmentirosa($stack,$fecha,'ibmMvsTcpListenerLocalPort.999
997 : INTEGER: 4209','999997','10.99.8.4','4209');
            }
            break;
        case("10.99.8.4.4210"):
            if ($dato == "established") {

procesoconexionmentirosa($stack,$fecha,'ibmMvsTcpListenerLocalPort.999
996 : INTEGER: 4210','999996','10.99.8.4','4210');
            }
            break;
        default:
            switch($dato)
            {
                case("listen"):
                    break;
                case("established"):

$matriz_established[$p5]=$matriz_established[$p5]+1;
                    break;
                case("finWait2"):
                    $matriz_finWait2[$p5]=$matriz_finWait2[$p5]+1;
                    break;
                case("timeWait"):
                    $matriz_timeWait[$p5]=$matriz_timeWait[$p5]+1;
                    break;
                case("closeWait"):

```

```
        $matriz_closeWait[$p5]=$matriz_closeWait[$p5]+1;
        break;
    default:
        break;
    }
    break;
}
}

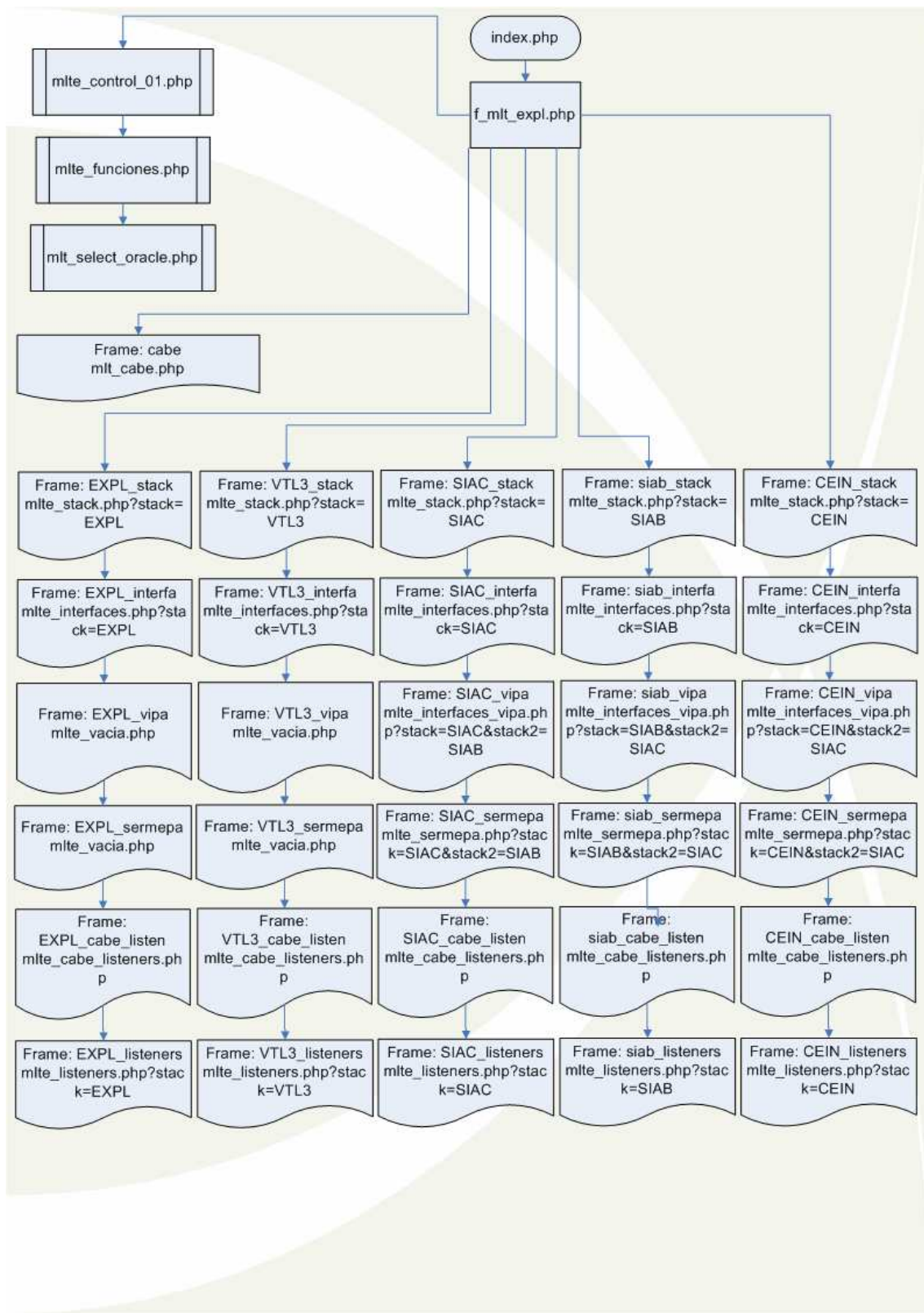
function ejecutar_insert($sentencia,$debug) {
    if ($debug == 1) {
        $hora = date("F j, Y, g:i a");
        echo $hora;
        echo "--".$sentencia;
        echo "sentencia:".$sentencia;
        echo "return code:".$ret_code;
        echo "Mensaje".$Msg;
    }
    $Msg="";
    $ret_code=exec_sql_unlogon($sentencia, $Msg);
    if ($ret_code != 0) {
        $hora = date("F j, Y, g:i a");
        echo "Hora: ".$hora."\n";
        echo "sentencia: ".$sentencia."\n";
        echo "return code: ".$ret_code."\n";
        echo "Mensaje ".$Msg."\n"; }
}

function exec_sql_unlogon($sentencia,&$Msg)
{
    if ($curs=@oci_parse($_SESSION['conn'], $sentencia))
    {if (@oci_execute($curs))
        { $Result = 0;
        }
        else
        {
            $e = oci_error($curs);
            $Msg=$e['message'];
            $_SESSION['mensaje'] = $Msg;
            $_SESSION['resultado_oracle'] = substr($Msg,4,5);
            $Result = substr($Msg,4,5);
        } // fin del oci_execute
        @oci_free_statement($curs);
    }
    else
    {
        $e = oci_error($curs);
        $Msg=$e['message'];
        $_SESSION['mensaje'] = $Msg;
        $_SESSION['resultado_oracle'] = substr($Msg,4,5);
        $Result = substr($Msg,4,5);
    } // fin del oci_parse

    return $Result;
}

?>
```

## V.4 Presentación de los Datos (Diagrama de flujo)



## V.5 Programas

### index.php

Esta es la página inicial desde donde se llama al programa principal.

### f\_mlt\_expl.php

Programa principal donde se definen los distintos frames y desde donde se realiza el refresco automático cada 60 segundos. Se decide la utilización de frames ya que me permite la inclusión/exclusión de nuevas particiones de una forma sencilla y rápida.

### mlte\_control\_01.php

Este es el programa más importante de todos ya que es el encargado de realizar las consultas a la base de datos y de rellenar las variables globales que luego se utilizarán en el resto de programas que representan los datos. Para ello utiliza **mlte\_funciones.php** y **mlt\_select\_oracle.php**.

### mlte\_funciones.php

En este programa están definidas las funciones control, datos\_del\_stack, interfaces\_fisicos, interfaces\_vipa, vipa\_port, sermepa y listeners entre otras que realizan las consultas a la base de datos.

### mlt\_select\_oracle.php

Este programa es el que se utiliza para acceder a la base de datos de forma centralizada y controlada.

### mlt\_cabe.php

Este programa representa la cabecera del monitor (icono, titulo, fecha, hora)

### mlte\_stack.php

Este programa representa los datos del stack (nombre de la particion, alarma activada, fecha de los datos, sockets establecidos y sockets totales). En el caso de que el número de sockets totales exceda un umbral se pone en amarillo y activa el

sistema de alarmas. Pinchando sobre el nombre de la partición nos da información adicional e histórica con una profundidad máxima de 3 días. Para ello utiliza el programa **f\_mlte\_stack.php**.

### **mlte\_interfaces.php**

Este programa representa los datos de los interfaces físicos . En el caso de que alguno de los interfaces este inoperativo se pone en rojo y activa el sistemas de alarmas. Pinchando sobre el nombre de los interfaces nos da información adicional (bytes de entrada, bytes de salida, errores, descartes, etc) e histórica con una profundidad máxima de 3 días. Parte de esta información se presenta de forma gráfica para su posible utilización en la realización de informes. Para ello utiliza el programa **f\_mlte\_porhoras\_int.php**.

### **mlte\_interfaces\_vipa.php**

Este programa representa los datos de los interfaces virtuales así como de su utilización. Estos interfaces virtuales solo están definidos en las particiones SIAC y SIAB. Mediante un código de colores sabemos en todo momento donde están activos esos interfaces y la utilización que se está haciendo de ellos.

### **mlte\_sermepa.php**

Este programa representa los datos de las conexiones con SERMEPA. Estas conexiones solo están definidas en las particiones SIAC y SIAB. Mediante un código de colores sabemos en todo momento donde están activas esas conexiones.

### **mlte\_vacia.php**

Este programa se utiliza para representar en vacio aquellos frames que no se utilizan en algunas particiones.

### **mlte\_cabe\_listeners.php**

Este programa representa la cabecera de los listeners.

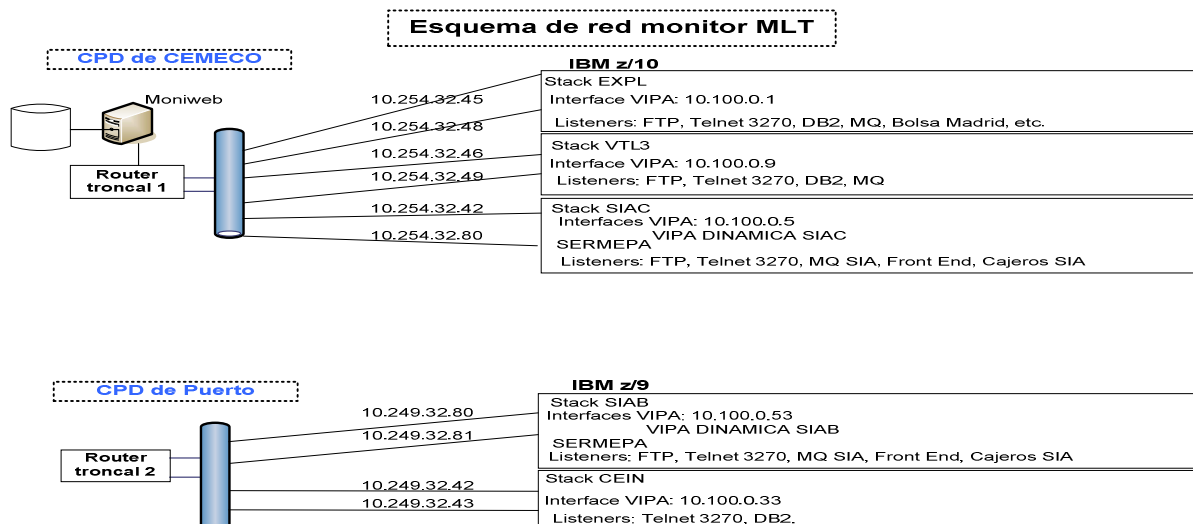
## mlte\_listeners.php

Este programa representa el estado de los listeners. En el caso de que alguno de los listeners no este operativo se pone en rojo y activa el sistema de alarmas. Pinchando sobre el nombre de los listeners nos da información adicional (número de conexiones established, accepts, closewait, finwait2, timewait, current backlog, exceeded backlog, max backlog) e histórica con una profundidad máxima de 3 días. Parte de esta información se presenta de forma gráfica para su posible utilización en la realización de informes. Para ello utiliza el programa **f\_mlte\_porhoras.php**.



## CAPITULO VI. MANUAL DE USUARIO

El Monitor Local de TCP/IP (en adelante, MLT), es un monitor de control del TCP/IP de las particiones EXPL, VTL3, SIAC, SIAB y CEIN del IBM. El panel principal presenta información de los STACK's TCP/IP del IBM (IP1EPROC, IP1VPROC, IP1CPROC, IP1BPROC, IP1NPROC) de los interfaces y de los LISTENERS de dicha partición.



La pantalla que presenta el programa es la siguiente:

MLT - Monitor Local de Comunicaciones IBM - Microsoft Internet Explorer proporcionado por Grupo Bancaja

Técnica de  
Sistemas  
Monitorización

Monitor Local TCP/IP (IBM) [09/Jul]

11:38

3862.5 - Técnica de Sistemas

EXPL

11:37:00

Total / Delta

Estab Sockets 4334 / -107

Total Sockets 4397 / -118

INTERFACES

VIPA - 10.100.0.1

10.254.32.45

10.254.32.48

VTL3

11:37:00

Total / Delta

Estab Sockets 25 / -1

Total Sockets 45 / -3

INTERFACES

VIPA - 10.100.0.9

10.254.32.46

10.254.32.49

SIAC

11:37:00

Total / Delta

Estab Sockets 28 / 0

Total Sockets 44 / 2

INTERFACES

VIPA - 10.100.0.5

10.254.32.42

10.254.32.80

VIPA S DINAMICAS

SIAC

SIAB

SIAC

SIAB

MQ

0

0

0

0

Front

0

0

0

0

Caj

545

0

0

0

SERMEPA

01

02

03

04

05

Emission

Recepcion

SIAB

11:37:00

Total / Delta

Estab Sockets 18 / 2

Total Sockets 40 / 0

INTERFACES

VIPA - 10.100.0.53

10.249.32.80

10.249.32.81

VIPA S DINAMICAS

SIAB

SIAC

SIAB

SIAC

MQ

0

0

0

0

Front

0

0

0

0

Caj

538

0

0

0

SERMEPA

01

02

03

04

05

Emission

Recepcion

CEIN

11:37:00

Total / Delta

Estab Sockets 174 / 7

Total Sockets 180 / 7

INTERFACES

VIPA - 10.100.0.33

10.249.32.42

10.249.32.43

LISTENERS

FTP

Telnet 3270

DB2 (Nóminas)

MQ Producción

Routing Bolsa Madrid (S)

Routing Bolsa Madrid (E)

Cicplex en web

Infobolsa

DB2 Resino (Nóminas)

DB2 (TIGRECAT)

DB2 Resino (TIGRECAT)

CFT

Editran

Gestion Documental (NPO)

Gestion Documental (BP)

Bancaja Proxima

V50 - TPTHSP

V50 - TPTHTP

Monitorización V50

LISTENERS

FTP

Telnet 3270

MQ

DB2 (M.Broker)

DB2 Resino (M. Broker)

MQ (M.Broker)

LISTENERS

FTP

Telnet 3270

Telnet 3270 SSL

MQ SIA

Front End

Cajeros SIA

LISTENERS

FTP

Telnet 3270

Telnet 3270 SSL

MQ SIA

Front End

Cajeros SIA

LISTENERS

Telnet 3270

DB2

DB2 Resino

Listo

Intranet local

A continuación detallaremos el manejo del monitor:

## 1. Requisitos para ejecutar el monitor

Es necesario:

- Un PC con la Plataforma de Oficinas Fase II o PC conectado a través de VPN con Internet Explorer

Es recomendable:

- Tarjeta de sonido instalada con drivers y altavoces
- Conexión a la red local

## 2. ¿Cómo se ejecuta MLT?

Para ejecutar el MLT, basta con acceder a:

<http://moniweb/monitores> y una vez logado a moniweb, pulsar sobre el icono del monitor:



## 3. Características principales

- Las alarmas sonoras se puede desactivar independientemente unas de otras.
- Si se dispone de tarjeta de sonido, las alarmas utilizan sonidos pregrabados en formato WAV. Si no se dispone de tarjeta, se escuchará un *beep* por el altavoz interno del PC.
- El código de colores de las alarmas es el siguiente:





**Rojo:** Recurso (interface, conexión) caído (Error).

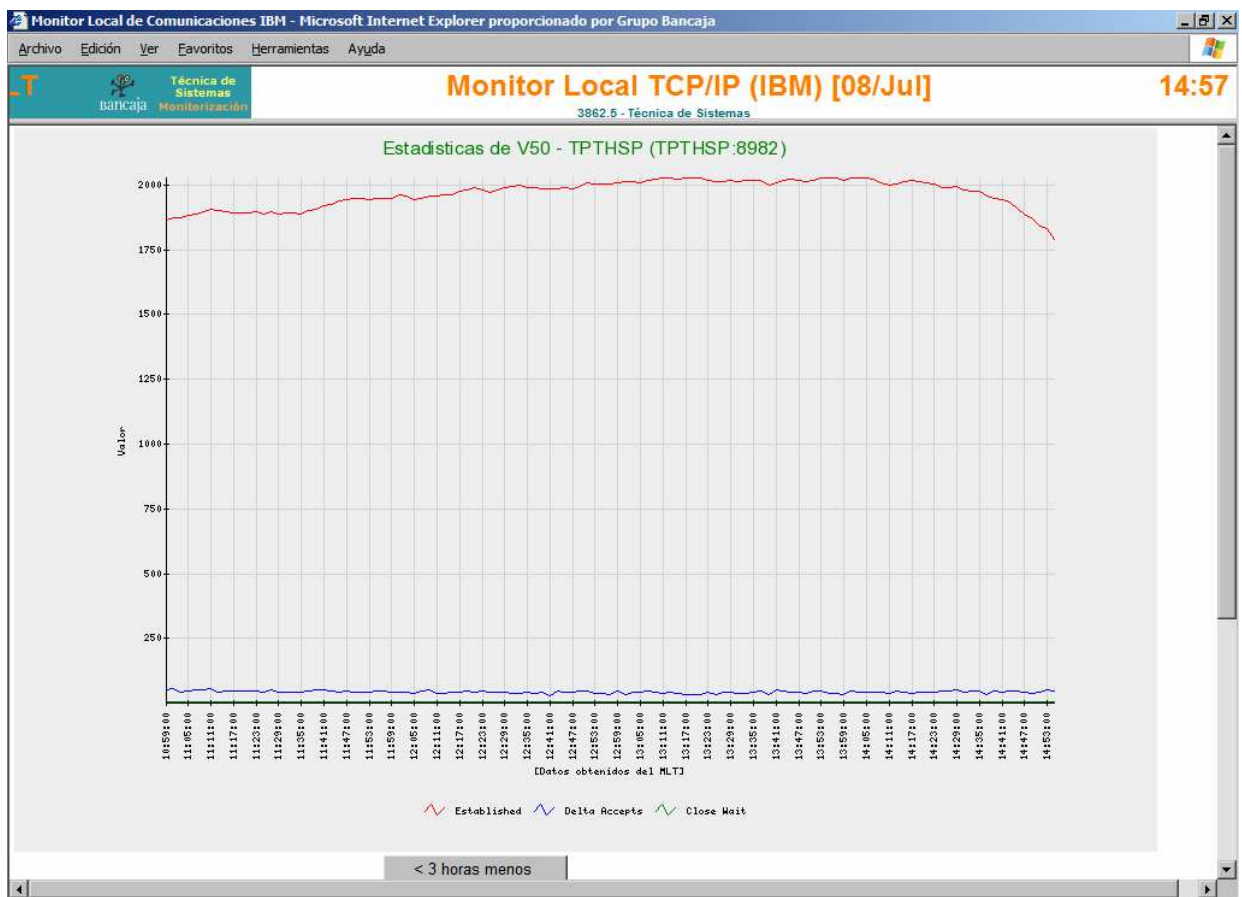
**Amarillo:** Posible problema.

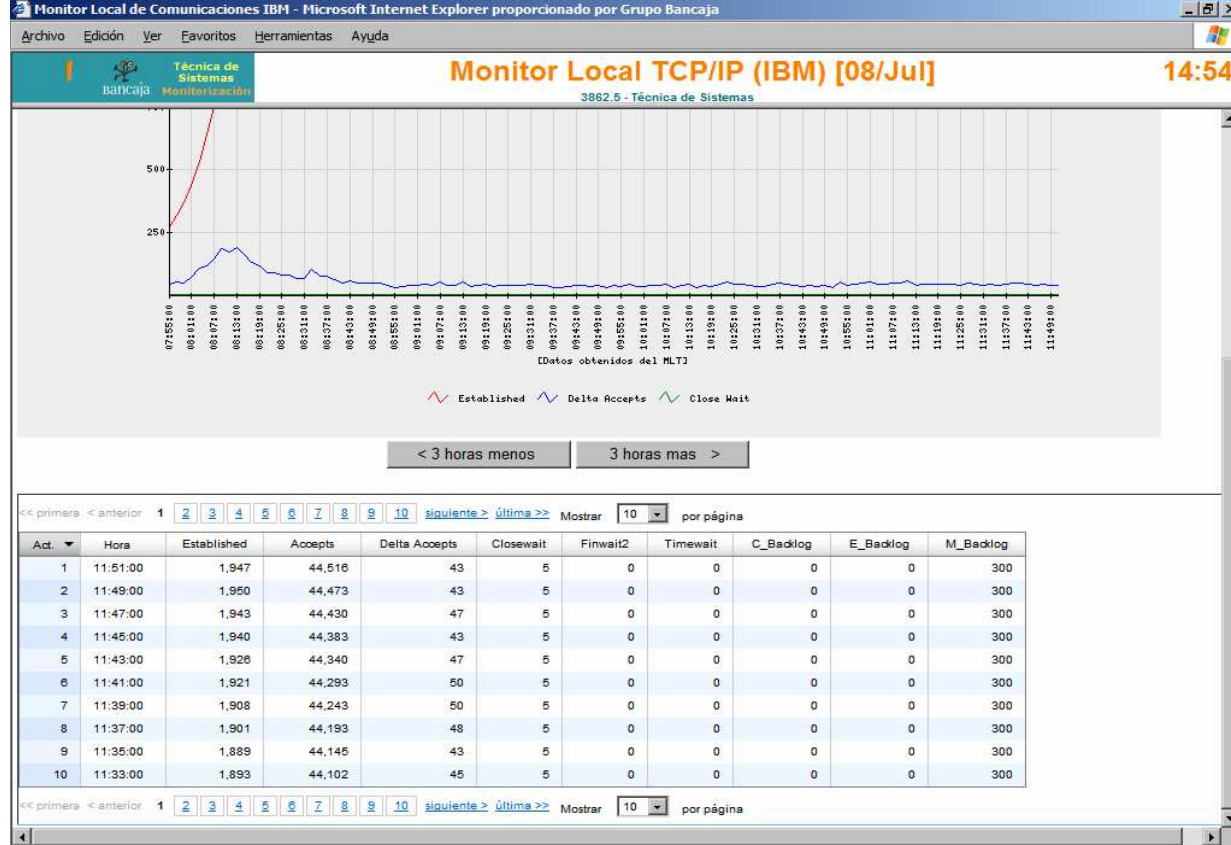
**Blanco:** Recurso (interface, conexión) activo.

**Gris:** Activo en otra partición (este color solo aparece en las conexiones de SERMEPA-SIA).

- Todos los campos de datos tiene ayudas, que se muestran simplemente dejando el cursor encima de dicho campo. Las ayudas aparecen en la parte inferior del monitor.

- Los botones con una marca de check (  ), campana (  ) o campana desactivada (  ) son los que tienen alarma. En el primer caso, no se está produciendo esa situación de error, en el segundo, sí que existe dicha situación y está sonando la alarma sonora. El tercer caso quita la alarma sonora. Para cambiar de los dos primeros estados al tercero y viceversa, basta con pinchar en el botón correspondiente o en la etiqueta asociada. Cuando la alarma está desactivada, la etiqueta también se muestra en color inverso.
- Se puede obtener una gráfica de los últimos valores en todos aquellos campos que al pasar por encima aparezca (  ). La página web, con la gráfica que se presentará, será similar a la siguiente:





#### 4. Uso del MLT

- **Activar/Desactivar Alarma:** Por defecto, cada vez que se arranca el MLT, todas las alarmas se encuentran activadas. Para cambiar su estado basta con pinchar con el ratón en la casilla dónde está el error. La desactivación de una alarma sólo afecta al sonido, pero los datos siguen refrescándose y se sigue manteniendo el código de colores.

#### 5. Descripción de la pantalla

A continuación veamos los grandes bloques que forman el monitor MLT:

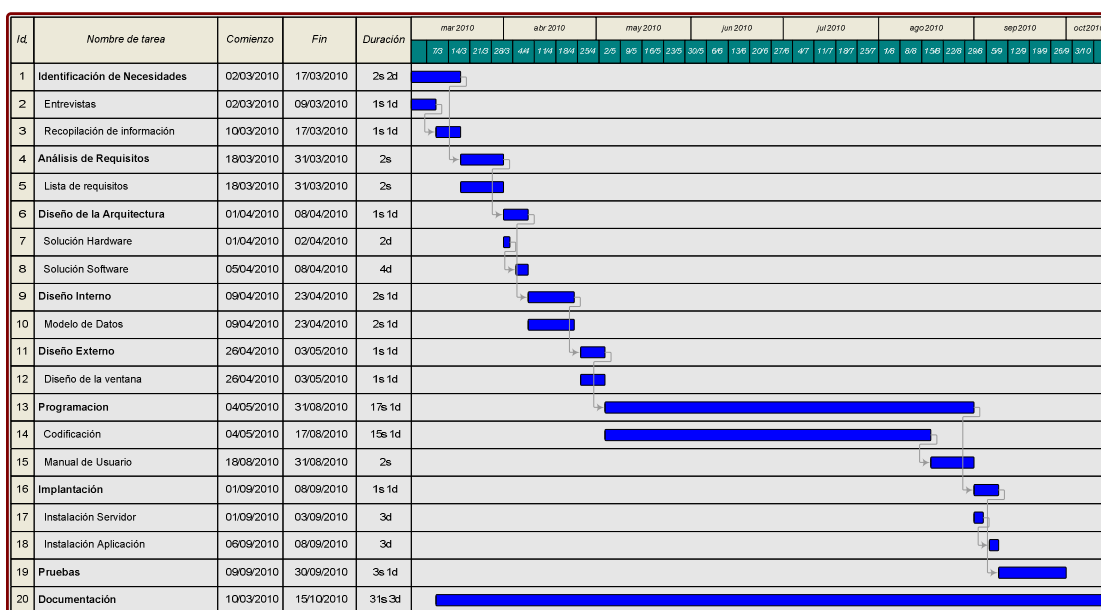
- **Nombre del STACK:** Proporciona información relativa del STACK (Curr Sockets, Total Sockets.). Además aparece la hora de la última actualización de todos los datos del Monitor en ORACLE.
- **Interfaces:** Proporciona información relativa acerca del estado de la VIPA estática y de sus interfaces físicas. Pinchando sobre el campo obtenemos una gráfica con los 60 últimos valores más significativos.
- **VIPAS DINAMICAS:** Estos aparecen en aquellas particiones que tienen VIPAS dinámicas. Proporciona información relativa acerca de su estado, de su utilización y de su ubicación actual.

- **SERMEPA:** Este apartado aparece en las particiones de SIA e indica el estado de los sockets de las distintas aplicaciones de SERMEPA. Si está en blanco significa que el socket está establecido pero no se han intercambiado datos.
- **Listeners:** Proporciona información relativa de los Listeners y Conexiones que se ha decidido monitorizar (Accepts, Conexiones, Backlog's).

## CAPITULO VII. PLANIFICACION TEMPORAL

La distribución del tiempo a lo largo del proyecto para cada una de las etapas ha sido la siguiente:

- Identificación de necesidades: Se ha recopilado información para familiarizarse con las tecnologías, herramientas y términos que se han utilizado a lo largo del proyecto. Se han estudiado las necesidades del cliente y se han realizado entrevistas con diferentes tipos de usuarios potenciales.
- Análisis de requisitos: Durante esta etapa se han analizado y documentado las necesidades funcionales o de servicio soportadas por el sistema propuesto. También se han identificado las restricciones y los requisitos no funcionales.
- Diseño de la arquitectura: Se ha determinado la arquitectura tecnológica de todo el sistema, tanto a nivel de software como de hardware y comunicaciones.
- Diseño interno y externo: En esta etapa se ha diseñado la interfaz gráfica de usuario, la base de datos, etc.
- Programación: Se ha escrito el código de la aplicación siguiendo lo establecido por los modelos de las etapas anteriores.
- Implantación.
- Pruebas: Se han realizado pruebas para certificar que la aplicación funciona de manera correcta.
- Documentación.



## CAPITULO VIII.BIBLIOGRAFIA

Durante el desarrollo del proyecto se ha utilizado una amplia cantidad de material bibliográfico. A continuación se muestran algunas de las fuentes que se han consultado en algún momento del proyecto.

- [1] M. G. SOBELL, Manual práctico de Linux, Ed. Anaya. 2005
- [2] *Manual on-line de Apache 1.3.* <http://httpd.apache.org/docs/>
- [3] *Oracle, sistema gestor de bases de datos* <http://www.oracle.com>
- [4] *mSQL, sistema gestor de bases de datos.* <http://www.hughes.com.au>
- [5] *MySQL, sistema gestor de bases de datos.* <http://www.mysql.com/>
- [6] *PostgreSQL, sistema gestor de bases de datos* <http://www.postgresql.org/>.
- [7] Enrique Rivero, Luis Martínez, Luis Reina, Juan Benavides, Juan M<sup>a</sup> Olaizola "Introducción al SQL para usuarios y programadores". Thomson, 2002.
- [8] Francisco Charte Ojeda, "PHP 5" Anaya Multimedia, 2004.
- [9] Página oficial de PHP, "<http://www.php.net>".
- [10] Página programación en PHP, "<http://www.programacion.com/php/>".
- [11] Leo Juszkievicz, Manual de HTML en <http://www.lawebdelprogramador.com> 2005.
- [12] Manual de HTML, <http://www.webestilo.com/html/>
- [13] Guía breve de CSS, <http://www.w3c.es/divulgacion/guiasbreves/HojasEstilo>
- [14] Editor de texto VI, <http://www.dc.fi.udc.es/~afyanez/info-vi/index.html>.
- [15] Bourne-Again Shell Home Page, <http://www.gnu.org/software/bash/bash.html>.